

**500  
POKES  
PEEKS 'N EXECS  
FOR THE  
TRS - 80 COLOR COMPUTER**

**Kishore M. Santwani**

**500  
POKES  
PEEKS 'N EXECs  
FOR THE  
TRS - 80 COLOR COMPUTER**

*SUPPLIED BY*  
**PARIS RADIO ELECTRONICS**  
161 BUNNERSONG RD CHIROSFIELD 7032  
P.O. BOX 380 DARLINGHURST 2010  
P/L 344-9111

**Kishore M. Santwani**

500  
POKES  
PEEKS 'N EXECs  
for the  
TRS-80 COLOR COMPUTER

by  
Kishore M. Santwani

Copyright (c) 1984, 1985, 1986 by MICROCOM SOFTWARE  
P.O. BOX 214  
FAIRPORT, N.Y. 14450  
PH: (716) 223-1477

**ALL RIGHTS RESERVED.** The text of this publication, or any part thereof, may not be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, downloading, storage in an information retrieval system, or otherwise, without the written permission of the publisher.

Printed in the United States of America

First Printing July 1984  
Second Printing January 1985  
Third Printing September 1985  
Fourth Printing April 1986

Every effort has been made to insure the accuracy of the contents of this book. Microcom Software does not guarantee the accuracy of or the information contained in this book, nor does it assume any liability arising out of the application or use of the information herein.

TRS-80 is a trade mark of the Tandy Corporation.

## CONTENTS

Preface .....	v
How To Use The Book .....	vi
FOR CASSETTE AND DISK SYSTEMS .....	1
Maximum Memory .....	1
Basic Program Merge .....	2
Reset Button disable .....	5
ML Exec Address .....	8
Start/End address of hi-res screen .....	11
X/Y values in hi-res screen .....	12
Slowing keyboard response .....	14
All Color Basic/ECB/Disk Basic	
Commands/Functions disables .....	15
ECB DEFUSR/USR Vector Address .....	15
Keyboard Rollover bits .....	17
Slowing down printing rate .....	18
Using Graphics without returning	
to text screen .....	18
Disabling all keys .....	18
Double-Print Mode .....	19
Cassette I/O .....	20
Determining Color Basic/ECB version .....	23
Substitute for 'INKEY\$' .....	26
Turning on/off motor relay .....	28
Rompak auto-execute disable .....	29
Hi-speed pokes .....	29
Graphic/Semigraphic modes .....	30
Utility Routines .....	35
Color Basic Commands disables .....	35
ECB Commands disables .....	37
Screen-to-printer mode .....	39
Reset,Motor on/off from keyboard .....	40
Key Disables .....	41
Break-key Disable .....	42

Key-Beep .....	42
Error-Beep .....	43
Starting Basic Programs with the RESET button .....	43
Recovering lost BASIC program .....	44
Repeat Key .....	44
Autostarting Basic programs .....	44
 Printer .....	45
Setting baud rates .....	45
line return delay .....	47
line width .....	47
Disk directory to printer .....	48
Determining printer status .....	48
 FOR DISK SYSTEM ONLY .....	49
Maximum Memory .....	49
Determining Disk Errors .....	50
All Disk Commands disables .....	50
Disk File Types .....	51
Disk DEFUSR/USR Vectors .....	53
Disk Commands disables .....	58
 FOR 64K CASSETTE & DISK SYSTEMS .....	61
Setting 64K All Ram Mode .....	61
Setting the Computer to Color Basic .....	61
Changing Cursor Prompts .....	61
Key Disables .....	62
Autostarting Basic Programs .....	63
Changing/Disabling OK prompt .....	63
Break-key disable .....	63
Setting computer to ECB .....	64
ROMPAK TRANSFER TO TAPE .....	64
 GAMES .....	65
Increasing the number of plays in 23 different arcade games .....	65

## P R E F A C E

500 POKEs PEEKs 'N EXECs FOR THE TRS-80 COLOR COMPUTER is a unique book inasmuch as it contains information which is of utmost significance to any programmer. This information is generally not found in one place and the people who know are hesitant to divulge this valuable and confidential information. It has therefore taken a great deal of research to prepare this book.

This book will unravel the POKE, PEEK and EXEC commands of the Color Computer to help you gain the power of Assembly Language thru Basic. It includes various commands relating to almost all the aspects of the Color Computer programming. In order to help you reach greater heights in successful programming, a number of Utility Routines like Autostarting your Basic programs from cassette, Disabling various Color/ECB/Disk Basic commands, and Key- and Error-beep have been included.

Besides the utilization of the basic knowledge of the inner working of the Color Computer for creation of various POKE, PEEK and EXEC commands, help has also been derived from various books and journals in the preparation of this book.

We shall be glad to receive your views on the book and any changes/additions to be incorporated to make it even more useful.

GOOD LUCK!!!

## HOW TO USE THE BOOK

500 POKEs PEEKs 'N EXECs FOR THE TRS-80 COLOR COMPUTER has been prepared in a style that can be easily followed by a neophyte as well as a professional programmer. It follows a logical sequence of major groups based on the different Systems — Cassette and Disk Systems. Disk System only, Printer, and 64K Cassette & Disk Systems.

Each POKE, PEEK & EXEC command is followed by the ensuing RESULT and detailed REMARKS to explain any additional features or to caution about the likely effects of the command. An exhaustive Table of Contents has been included to help you locate the major commands in the book.

### USE OF POKE, PEEK AND EXEC COMMANDS

The Poke, Peek & Exec Commands are used as follows:

**POKE:** The Poke command is used independently. It comprises two parts, namely, the address to be poked into and the value to be poked into the address. The format for Poke is 'POKE xx.yy', where xx = address; yy = value to be poked into xx. To use a Poke command, simply type in the command and press the [ENTER] key.

**PEEK:** The Peek command is used to retrieve a value from a location. It is always preceded by a variable. The format for Peek is 'X = PEEK(xx)' where X is ANY variable and xx is the memory location from where the value is to be retrieved. The retrieved value is then assigned to variable 'X'.

**EXEC:** The Exec command is used to JUMP to a set of instructions in memory. The format for 'EXEC' is 'EXEC xx' where xx is the location to be JUMPed to. To use the Exec command simply type in the command and press the [ENTER] key.

## FOR CASSETTE AND DISK SYSTEMS

Most of the commands listed under this section are compatible with 16K/32K/64K Cassette and Disk Systems. However certain commands are not compatible with disk systems and are so mentioned in the REMARKS. Similarly, most commands are compatible with both Color Basic and ECB, while some are exclusively for Color Basic OR ECB OR Disk Basic. These are specifically mentioned in the RESULT or the REMARKS.

### Utility Routines

One of the important features of this book is to help you utilize various POKE, PEEK and EXEC commands in different utility functions. These utility routines should have a statement number and should be placed in the beginning of your Basic program.

Certain Color and ECB commands disables have been included under the utility routines (Disk Commands Disables are listed under the section 'FOR DISK SYSTEM ONLY'). In order to disable various Color Basic/ECB/Disk Commands, first type in the BASE Command Pokes. The BASE Command Pokes for Color Basic/ECB/Disk Basic are given in the BEGINNING of each group and are followed by individual pokes which disable a SINGLE command. Once you have typed in the BASE pokes for a group (Color Basic/ECB/Disk Basic), you may use ONE or MORE of the individual pokes to disable various commands. NOTE: DO NOT REPEAT THE BASE POKES FOR EACH INDIVIDUAL COMMAND DISABLE POKE. The same applies to the KEY-DISABLEs, except that you may use only ONE of the individual Key-Disable pokes following the BASE pokes.

### FOR DISK SYSTEM ONLY

All Commands listed under this section require 16K/32K/64K Disk System and are only compatible with ECB.

## FOR 64K CASSETTE AND DISK SYSTEMS

All commands listed under this section require a 64K Color Computer and the 64K ALL RAM MODE given in the beginning of the section. The command to set the ALL RAM MODE should have a statement number and should be 'RUN' to set the 64K ALL RAM MODE. The command to set the ALL RAM MODE should preferably be given in the beginning of your Basic program.

## GAMES

The commands listed under this section help you to get more plays on 23 different arcade games. The individual commands should be typed in AFTER '(C)LOADM'ing the arcade-game and BEFORE typing 'EXEC'. The names of the various arcade-games used in this section are Trademarks of different companies, and have been included for the sake of easy comprehension.

## ABBREVIATIONS

Certain abbreviations have been used in this book which are given below:

ABBREVIATION	EXPANDED FORM
Coord.	Coordinate
CSS	Color Set
ECB	Extended Basic
For ex.	For example
I/O	Input/Output
Req.	Required
X = 1 - 255	X = 1 to 255.

## FOR CASSETTE & DISK SYSTEMS

**Command:** PEEK(0)

**Result:** Returns a number < 128 if a basic program is halted by an 'END' statement or >=128 if halted by a 'STOP' statement.

**Remarks:** None

**Command:** PEEK(9)

**Result:** Returns a < > 0 number if data entered by a 'READ' statement or 0 if data entered by an 'INPUT' statement.

**Remarks:** None

**Command:** PEEK(25)\*256 + PEEK(26)

**Result:** Returns the starting address of a basic program in memory.

**Remarks:** None

**Command:** POKE 25,6:POKE 26,1:POKE 1536,0:NEW

**Result:** Makes maximum memory available for basic programs.

**Remarks:** Any basic program in memory is cleared. Not compatible with disk systems. Does not allow the use of graphics.

**Command:** POKE 25,12:POKE 26,1:POKE 3072,0:NEW

**Result:** Sets basic program starting address at the 'PCLEAR 1' page.

**Remarks:** Not compatible with the disk system. Allows limited use of graphics. Clears any basic program in memory.

**Command:** POKE 25,18:POKE 26,1:POKE 4608,0:NEW

**Result:** Sets basic program starting address at the 'PCLEAR 2' page.

**Remarks:** Not compatible with disk systems. Allows limited use of graphics. Clears any basic program in memory.

**Command:** POKE 25.24:POKE 26,1:POKE 6144,0:NEW  
**Result:** Sets Basic program starting address at the 'PCLEAR 3' Page.  
**Remarks:** Not compatible with Disk Systems. Allows limited use of graphics. Clears any basic program in memory.

**Command:** POKE 25.30:POKE 26,1:POKE 7680,0:NEW  
**Result:** Sets Basic program starting address at the 'PCLEAR 4' Page.  
**Remarks:** Not compatible with Disk Systems. Clears any basic program in memory.

**Command:** [CLOAD"PROGRAM1"]:POKE 118,PEEK(25):POKE 119,PEEK(26):X = (PEEK(27) \* 256 + PEEK(28)) - 2: A = INT(X/256):POKE 25,A:POKE 26,(X - (A \* 256)):NEW :[CLOAD"PROGRAM2"]:POKE 25,PEEK(118):POKE 26,PEEK(119)  
**Result:** Merges "PROGRAM1" and "PROGRAM2".  
**Remarks:** Statement numbers of these two programs should not overlap. Merges Basic programs only.

**Command:** PEEK(27) \* 256 + PEEK(28)  
**Result:** Returns the ending address of the Basic program in memory.  
**Remarks:** Variable storage starts at this address.

**Command:** (PEEK(27) \* 256 + PEEK(28)) - (PEEK(25) \* 256 + PEEK(26))  
**Result:** Returns the number of bytes in a Basic program in memory.  
**Remarks:** Number of bytes = ending address - starting address.

**Command:** PEEK(29) \* 256 + PEEK(30)  
**Result:** Returns the starting address of array storage.  
**Remarks:** None.

**Command:** (PEEK(31) \* 256 + PEEK(32)) + 1  
**Result:** Returns the ending address of array storage.  
**Remarks:** None.

**Command:** PEEK(33)\*256+PEEK(34)  
**Result:** Returns the address of top of free RAM (start of string storage).  
**Remarks:** The top of free RAM can be changed by the 'CLEAR xx' command.

**Command:** (PEEK(35)\*256+PEEK(36))+1  
**Result:** Shows place of protected memory.  
**Remarks:** For example if the protected memory starts at 20000 which is set by 'CLEAR 10,20000', the number returned by the above statement equals 20000.

**Command:** (PEEK(35)\*256+PEEK(36))-(PEEK(33)\*256+PEEK(34))  
**Result:** Returns the amount of string space reserved by 'CLEAR xx'.  
**Remarks:** For example if string space is reserved by 'CLEAR 2000' then the above command returns the value of 2000.

**Command:** (PEEK(39)\*256+PEEK(40))+1  
**Result:** Shows place of protected memory.  
**Remarks:** Same as in (PEEK(35)\*256+PEEK(36))+1.

**Command:** PEEK(41)\*256+PEEK(42)  
**Result:** Returns the statement number where a Basic program was halted by a 'STOP' command or the [BREAK KEY].  
**Remarks:** None.

**Command:** PEEK(43)\*256+PEEK(44)  
**Result:** Returns the most recent statement number executed.  
**Remarks:** In case an error occurred within a Basic program, it returns the error line number.

**Command:** POKE 45,0:POKE 46,0  
**Result:** Disables 'CONT' Function after break.  
**Remarks:** 'CONT' command after these POKEs produces a '?CN ERROR'.

**Command:** PEEK(49)\*256+PEEK(50)  
**Result:** Returns current data line number.  
**Remarks:** None.

**Command:** CHR\$(PEEK(55))+CHR\$(PEEK(56))  
**Result:** Last numeric variable used in a basic program.  
**Remarks:** None.

**Command:** CHR\$(PEEK(55))+CHR\$(PEEK(56)-128)  
**Result:** Returns last string variable used in a basic program.  
**Remarks:** None.

**Command:** PEEK(86)\*256+PEEK(87)  
**Result:** Returns current address during 'LIST'.  
**Remarks:** None.

**Command:** PEEK(104)\*256+PEEK(105)  
**Result:** Current line number of basic program.  
**Remarks:** Returns 65535 if this command is used in Direct Mode.

**Command:** POKE 104,0:POKE 105,0:CLOAD  
**Result:** Loads a basic program from cassette without blinking a 'S' or a 'F' in the top left corner of the screen.  
**Remarks:** These commands should be used from Direct Mode.

**Command:** POKE 104,255:POKE 105,255:CLOAD  
**Result:** Loads a basic program from cassette with a blinking 'S' or 'F'.  
**Remarks:** These commands should be used within a basic program.

**Command:** PEEK(106)  
**Result:** Returns comma column width.  
**Remarks:** None.

**Command:** PEEK(108)

**Result:** Returns current print position in line.

**Remarks:** Returns number of characters from the left margin of the screen, same as the 'POS(0)' Command of Extended Basic.

**Command:** POKE 113,0

**Result:** Disables the reset button.

**Remarks:** Computer coldstarts when the reset button is pressed.

**Command:** POKE 113,85

**Result:** Enables the reset button after POKE 113,0.

**Remarks:** Computer warmstarts when the reset button is pressed.

**Command:** POKE 113,0:EXEC 40999

**Result:** Produces an immediate coldstart.

**Remarks:** Erases 1K(1024 Bytes) of Ram Memory starting at address 0.

**Command:** PEEK(113)

**Result:** Tests if the computer is setup for a warmstart or a coldstart.

**Remarks:** Returns 85 if the computer is setup for a warmstart or a < 85 number if the computer is setup for a coldstart.

**Command:** PEEK(114)\*256 + PEEK(115)

**Result:** Returns jump address for warmstart.

**Remarks:** If PEEK(113) = 85 and  
PEEK(PEEK(114)\*256 + PEEK(115)) = 18 then basic will jump to this address when reset button is pressed.

**Command:** POKE 114,0:POKE 115,0

**Result:** Disables the reset button.

**Remarks:** Computer coldstarts when the reset button is pressed.

**Command:** PEEK(116)

**Result:** Returns 127 in a 32K/64K System or 63 in a 16K System.

**Remarks:** Can be used to ascertain the memory size of a computer.

**Command:** PEEK(116)\*256 + PEEK(117)

**Result:** Returns maximum memory in System.

**Remarks:** None.

**Command:** PEEK(120)

**Result:** Returns 2 if a Data File is opened to cassette for output.

**Remarks:** Data file is opened for input by: OPEN "0",#-1,"FILENAME".

**Command:** PEEK(120)

**Result:** Returns 1 if data file is opened to cassette for input.

**Remarks:** Data file is opened by: OPEN "1",#-1,"FILENAME".

**Command:** PEEK(120)

**Result:** Returns 0 if a data file is closed to cassette.

**Remarks:** Data file can be closed to cassette by: CLOSE #-1.

**Command:** PEEK(124)

**Result:** Returns 0 if a HEADER BLOCK is loaded from cassette.

**Remarks:** None.

**Command:** PEEK(124)

**Result:** Returns 1 if a DATA BLOCK is loaded from cassette.

**Remarks:** None.

**Command:** PEEK(124)

**Result:** Returns 255 if a EOF BLOCK is loaded from cassette.

**Remarks:** None.

**Command:** PEEK(126)\*256 + PEEK(127)-1

**Result:** Returns the end address of a ML Program loaded from cassette.

**Remarks:** Not compatible with disk programs.

**Command:** PEEK(126)\*256 + PEEK(127)

**Result:** Returns the address at which the data read from cassette is stored in the computer memory.

**Remarks:** None.

**Command:** PEEK(129)

**Result:** Returns 2 if a cassette file is loaded into NONRAM.

**Remarks:** None.

**Command:** POKE 135,0

**Result:** Resets INKEY\$ Value to null.

**Remarks:** Resets the most recent INKEY\$ Value.

**Command:** (PEEK(136)\*256 + PEEK(137))—1024

**Result:** Returns cursor position in text screen.

**Remarks:** Cursor position is between 0 and 511.

**Command:** POKE 136,X:POKE 137,Y

**Result:** Sets cursor position in text screen.

**Remarks:** The above command sets the cursor position at  $(X * 256 + Y) - 1024$  location in the text screen.  
 $(X * 256 + Y)$  Must be  $> 1024$  &  $< 1535$ .

**Command:** POKE 136,X:POKE 137,Y:EXEC &HA1A1:GOTO xx.

**Result:** Flashes the cursor at text screen location  
 $(X * 256 + Y) - 1024$ .

**Remarks:** Cannot be used in Direct Mode. xx Denotes the Statement No. of the above command.  $(X * 256 + Y)$  Must be  $> 1024$  and  $< 1535$ .

**Command:** PEEK(140)

**Result:** Returns the tone value of the most recently executed 'SOUND xx,yy' Statement.

**Remarks:** None.

**Command:** POKE 140,X:POKE 141,Y:POKE 142,Z:EXEC 43350  
**Result:** Generates sound with tone X & Duration  
$$(Y * 256 + Z) / 4.$$
  
**Remarks:** To find the values of Y & Z -- A = DURATION \* 4:Y = INT(A/256):Z = (A - (Y \* 256)).

**Command:** POKE 148.0  
**Result:** Stops cursor flashing for approximately 3 seconds.  
**Remarks:** None.

**Command:** POKE 148.255  
**Result:** Stops cursor flashing for approximately 3 seconds.  
**Remarks:** None.

**Command:** POKE 148.11  
**Result:** Restore flashing cursor after POKE 148.0 or POKE 148.255.  
**Remarks:** None.

**Command:** PEEK(148)  
**Result:** Returns current cursor color.  
**Remarks:** Cursor color is between 1 and 11.

**Command:** PEEK(157)\*256 + PEEK(158)  
**Result:** Returns the EXEC ADDRESS of a disk or cassette loaded ML PRO.  
**Remarks:** None.

**Command:** POKE 157,178:POKE 158,119  
**Result:** Disables 'EXEC' Command.  
**Remarks:** Works only if 'EXEC' is not suffixed by an address.

**Command:** POKE 175,79  
**Result:** Turns on Trace Flag.  
**Remarks:** Same as the 'TRON' Command of Extended Basic.

**Command:** POKE 175.0

**Result:** Turns off Trace Flag.

**Remarks:** Same as the 'TROFF' command of Extended Basic.

**Command:** PEEK(175)

**Result:** Returns 0 if Trace Flag is off.

**Remarks:** Works only with Extended Basic.

**Command:** PEEK(175)

**Result:** Returns A < > 0 Number if Trace Flag is on.

**Remarks:** Works only with Extended Basic.

**Command:** POKE 178.X

**Result:** Produces various designs in commands like 'LINE', 'CIRCLE', etc.

**Remarks:** X = 1 to 255. ECB required. Try: PMODE3.1:PCLS: SCREEN1.1:POKE178.50:LINE (128.96)-(230.190), PSET.BF.

**Command:** PEEK(178) + 1

**Result:** Returns foreground color between 1 and 4.

**Remarks:** Requires ECB. Works only in four color graphic modes. Set by 'COLOR x,y' command.

**Command:** POKE 179.X:PCLS

**Result:** Produces various background designs in graphic modes.

**Remarks:** Requires ECB. X = 1-255. TRY: PMODE3.1:SCREEN 1.1:POKE179.10:PCLS.

**Command:** PEEK(179) + 1

**Result:** Returns background color between 1 and 4.

**Remarks:** See 'PEEK(178) + 1'.

**Command:** PEEK(181)

**Result:** Returns current color in 4 color graphic modes.

**Remarks:** Legal values for colors 1-4 are 0,85,170,255. Requires ECB.

**Command:** PEEK(181)  
**Result:** Returns current color in 2 color graphic modes.  
**Remarks:** Legal values for colors 1—4 are 255.0,255.0.  
Requires ECB.

**Command:** PEEK(182)  
**Result:** Returns current 'PMODE'.  
**Remarks:** Requires ECB.

**Command:** PEEK(183) \* 256 + PEEK(184)  
**Result:** Returns the end address of current HI—RES Screen.  
**Remarks:** Requires ECB.

**Command:** PEEK(185)  
**Result:** Returns the no. of bytes in a horizontal graphic line.  
**Remarks:** Requires ECB.

**Command:** PEEK(186) \* 256 + PEEK(187)  
**Result:** Returns start address of current HI—RES Screen.  
**Remarks:** Requires ECB.

**Command:** PEEK(188)  
**Result:** Returns 14 with a Disk System or 6 with a Cassette System.  
**Remarks:** Helpful to ascertain if disk or cassette system in operation.

**Command:** PEEK(189) \* 256 + PEEK(190)  
**Result:** Returns the starting X Coordinate. Requires ECB.  
**Remarks:** For ex. it will return 128 with LINE(128.96)—(230.100).PSET.

**Command:** PEEK(191) \* 256 + PEEK(192)  
**Result:** Returns the starting Y Coordinate. Requires ECB.  
**Remarks:** For ex. it will return 96 with LINE(128.96)—(230.100).PSET.

**Command:** PEEK(193)  
**Result:** Returns 8 with colorset 1 and 0 with colorset 0.  
**Remarks:** Requires ECB.

**Command:** PEEK(194)  
**Result:** Returns 1 if graphic pixel set by 'PSET' or 0 if by 'PRESET'.  
**Remarks:** Requires ECB.

**Command:** PEEK(195) \* 256 + PEEK(196)  
**Result:** Returns the ending X Coordinate value. Requires ECB.  
**Remarks:** For ex. it will return 230 with LINE(128,96)—(230,100),PSET.

**Command:** PEEK(197) \* 256 + PEEK(198)  
**Result:** Returns the ending Y Coordinate value. Requires ECB.  
**Remarks:** For ex. it will return 100 with LINE(128,96)—(230,100),PSET.

**Command:** PEEK(199) \* 256 + PEEK(200)  
**Result:** Returns previous X Coordinate value. Requires ECB.  
**Remarks:** For ex. it returns 230 with LINE(128,96)—(230,100),PSET. Sets the starting X Coord (230) for LINE—(250,120),PSET.

**Command:** PEEK(201) \* 256 + PEEK(202)  
**Result:** Returns previous Y Coordinate value. Requires ECB.  
**Remarks:** For ex. it returns 100 with LINE(128,96)—(230,100),PSET. Sets the starting Y Coord(100) for LINE—(250,120),PSET.

**Command:** PEEK(203) \* 256 + PEEK(204)  
**Result:** Returns circle X Center Coordinate value in PMODE 4. Req. ECB.  
**Remarks:**  $2 * (PEEK(203) * 256 + PEEK(204)) = \text{Circle Cent. X Coord in PMODES 0-3.}$

**Command:** PEEK(205) \* 256 + PEEK(206)  
**Result:** Returns Circle Y Center Coordinate value in PMODES 2—4. Req. ECB.  
**Remarks:**  $2 * (PEEK(205) * 256 + PEEK(206)) = \text{Circle Cent. Y Coord in PMODES 0-1.}$

**Command:** PEEK(207) • 256 + PEEK(208)  
**Result:** Returns circle radius in PMODE 4. Requires ECB.  
**Remarks:**  $2 \cdot (PEEK(207) \cdot 256 + PEEK(208)) = \text{Circle Radius in PMODES 0-3.}$

**Command:** PEEK(207) • 256 + PEEK(208)  
**Result:** Returns 'RENUM' Increment value.  
**Remarks:** For ex. it will return 5 in 'RENUM 10,15,5'. Requires ECB.

**Command:** (PEEK(209) • 256 + PEEK(210))/256  
**Result:** Returns the height/width ratio of a circle.  
**Remarks:** For ex. it returns 2 in 'CIRCLE(9.9),6,4,2,,5,,75'. Req. ECB.

**Command:** PEEK(209) • 256 + PEEK(210)  
**Result:** Returns 'RENUM' start (old) Line Number.  
**Remarks:** For ex. it returns 10 in 'RENUM 15,10,5'. Requires ECB.

**Command:** PEEK(213) • 256 + PEEK(214)  
**Result:** Returns 'RENUM' new line number.  
**Remarks:** For ex. it returns 15 in 'RENUM 15,10,5'. Requires ECB.

**Command:** PEEK(215) + 1  
**Result:** Returns No. of Characters in most recent 'EDIT'ED Statement.  
**Remarks:** For ex. it returns 7 in '5 Print A'. Requires ECB.

**Command:** PEEK(222) + 1  
**Result:** Returns current octave value used in the 'PLAY' command.  
**Remarks:** Octave = 1—5. For ex. it Returns 3 in 'PLAY"03L5A"'. Requires ECB.

**Command:** INT(PEEK(223)—PEEK(224))/8  
**Result:** Returns current volume value in the 'PLAY' command.  
**Remarks:** Volume = 1—31. For ex. it returns 6 in 'PLAY"V6L5A"''. Req. ECB.

**Command:** PEEK(225)  
**Result:** Returns current note length used in the 'PLAY' command  
**Remarks:** Note LEN. = 1—255. For ex. it Returns 5 in 'PLAY''L5A''. Req. ECB.

**Command:** PEEK(226)  
**Result:** Returns current tempo value used in the 'PLAY' command.  
**Remarks:** Tempo = 1—255. For ex. it returns 9 in 'PLAY''T9L25A''. Req. ECB.

**Command:** PEEK(230)  
**Result:** Returns 'DLOAD' Baud rate constant. Requires ECB.  
**Remarks:** Returns 44 with 'DLOAD' Baud Rate 1200 & 176 With Baud Rate 300.

**Command:** PEEK(232)  
**Result:** Returns current angle value used in the 'DRAW' command.  
**Remarks:** Angle = 1—3. For ex. it returns 2 with 'Draw''A2M9,9''. Requires ECB.

**Command:** PEEK(233)  
**Result:** Returns current scale value used in the 'DRAW' Command.  
**Remarks:** Scale = 1—62. For ex. it returns 8 in 'Draw''S8M9,9''. Req. ECB.

**Command:** PEEK(274) \* 256 + PEEK(275)  
**Result:** Returns current 'TIMER' Value.  
**Remarks:** Timer Value = 0—65535. Requires ECB.

**Command:** POKE 274,X:POKE 275,Y  
**Result:** Sets current timer value to X \* 256 + Y.  
**Remarks:** To find the values of X & Y —— A = Timer Value to be set: X = INT(A/256):Y = (A—(X \* 256)). Requires ECB.

**Command:** PEEK(274) \* 256 + PEEK(275)  
**Result:** Returns Jump Address for color basic 'USR' Function.  
**Remarks:** Compatible with Color Basic only.

**Command:** POKE 274,X:POKE 275,Y

**Result:** Sets 'USR' Jump Address for Color Basic.

**Remarks:** To find the values of X & Y -- A = 'USR' Value to be set: X = INT(A/256):Y = (A-(X \* 256)).  
Compatible with Color Basic only.

**Command:** POKE 280,PEEK(275)

**Result:** Helps COCO Generate 'TRUE' Random Numbers.

**Remarks:** Same as the Ext. Basic Statement --  
X = RND(-TIMER).

**Command:** POKE 282,0

**Result:** Turns on lowercase flag.

**Remarks:** Coco Generates lowercase characters.

**Command:** POKE 282,255

**Result:** Turns on uppercase flag.

**Remarks:** Coco Generates uppercase characters.

**Command:** PEEK(282)

**Result:** Returns 0 if lowercase flag is on.

**Remarks:** None.

**Command:** PEEK(282)

**Result:** Returns 255 if uppercase flag is on.

**Remarks:** None.

**Command:** POKE 283,X

**Result:** Slows down keyboard response.

**Remarks:** X = 5 to 255. 255 -- slowest.

**Command:** POKE 283,4

**Result:** Restores keyboard response to normal after the preceding poke.

**Remarks:** None.

**Command:** POKE 293,0  
**Result:** Disables all Color Basic & Extended Basic functions.  
**Remarks:** If Disk Basic is present, disk functions are also disabled.

**Command:** POKE 293,20  
**Result:** Enables Color Basic & ECB Functions disabled by POKE 293,0.  
**Remarks:** Also enables Disk Basic Functions if disk basic is present.

**Command:** POKE 306,178:POKE 307,119.  
**Result:** Disables all ECB & Disk Basic Functions.  
**Remarks:** None.

**Command:** POKE 306,129:POKE 307,104  
**Result:** Enables ECB & Disk Basic Functions after preceding command.  
**Remarks:** None.

**Command:** POKE 301,178:POKE 302,119  
**Result:** Disables all ECB & Disk Basic Commands.  
**Remarks:** None.

**Command:** POKE 301,129:POKE 302,60  
**Result:** Enables ECB & Disk Commands after 'POKE 301,18:POKE 302,119'.  
**Remarks:** None.

**Command:** PEEK(318)\*256 + PEEK(319)  
**Result:** Returns USR0 Vector for Extended Basic.  
**Remarks:** Set By --- 'DEFUSR0 = xxxx'. Not compatible with Disk System.

**Command:** PEEK(320)\*256 + PEEK(321)  
**Result:** Returns USR1 Vector for Extended Basic.  
**Remarks:** Set by --- 'DEFUSR1 = xxxx'. Not compatible with Disk System.

**Command:** PEEK(322)\*256 + PEEK(323)  
**Result:** Returns USR2 Vector for Extended Basic.  
**Remarks:** Set by —— 'DEFUSR2 = xxxx'. Not compatible with Disk System.

**Command:** PEEK(324)\*256 + PEEK(325)  
**Result:** Returns USR3 Vector for Extended Basic.  
**Remarks:** Set by —— 'DEFUSR3 = xxxx'. Not compatible with Disk System.

**Command:** PEEK(326)\*256 + PEEK(327)  
**Result:** Returns USR4 Vector for Extended Basic.  
**Remarks:** Set by —— 'DEFUSR4 = xxxx'. Not Compatible with Disk System.

**Command:** PEEK(328)\*256 + PEEK(329)  
**Result:** Returns USR5 Vector for Extended Basic.  
**Remarks:** Set by —— 'DEFUSR5 = xxxx'. Not compatible with Disk System.

**Command:** PEEK(330)\*256 + PEEK(331)  
**Result:** Returns USR6 Vector for Extended Basic.  
**Remarks:** Set by —— 'DEFUSR6 = xxxx'. Not compatible with Disk System.

**Command:** PEEK(332)\*256 + PEEK(333)  
**Result:** Returns USR7 Vector for Extended Basic.  
**Remarks:** Set by —— 'DEFUSR7 = xxxx'. Not compatible with Disk System.

**Command:** PEEK(334)\*256 + PEEK(335)  
**Result:** Returns USR8 Vector for Extended Basic.  
**Remarks:** Set by —— 'DEFUSR8 = xxxx'. Not compatible with Disk System.

**Command:** PEEK(336)\*256 + PEEK(337)  
**Result:** Returns USR9 Vector for Extended Basic.  
**Remarks:** Set by —— 'DEFUSR9 = xxxx'. Not compatible with Disk System.

**Command:** PEEK(338)

**Result:** Contains keyboard rollover bits @.H.P,X.0.8,  
[ENTER].

**Remarks:** Values of bits = 254,253,251,247,239,223,191  
Corresponding to the rollover bits. Returns 255 if  
no key is pressed.

**Command:** PEEK(339)

**Result:** Contains keyboard rollover bits A.I.Q,Y,1.9,  
[CLEAR].

**Remarks:** Refer to 'PEEK(338)' for bit values.

**Command:** PEEK(340)

**Result:** Contains keyboard rollover bits B.J.R.Z.2,:  
[BREAK].

**Remarks:** Refer to 'PEEK(338)' for bit values.

**Command:** PEEK(341)

**Result:** Contains keyboard rollover bits C.K.S.↑.3,:

**Remarks:** Refer to 'PEEK(338)' for bit values.

**Command:** PEEK(342)

**Result:** Contains keyboard rollover bits D.L.T.↓.4,comma.

**Remarks:** Refer to 'PEEK(338)' for bit values.

**Command:** PEEK(343)

**Result:** Contains keyboard rollover bits E.M.U.←.5,.

**Remarks:** Refer to 'PEEK(338)' for bit values.

**Command:** PEEK(344)

**Result:** Contains keyboard rollover bits  
F.N.V.→.6,period.

**Remarks:** Refer to 'PEEK(338)' for bit values.

**Command:** PEEK(345)

**Result:** Contains keyboard rollover bits G.O.W.space.7./.

**Remarks:** Refer to 'PEEK(338)' for bit values.

**Command:** POKE 359.60  
**Result:** Slows down character printing rate.  
**Remarks:** 'SCREEN 0.1' after this POKE produces an orange screen in ECB. Not compatible with Disk System.

**Command:** POKE 359.0  
**Result:** Restores normal character printing rate after 'POKE 359.60'.  
**Remarks:** Not compatible with Disk System.

**Command:** POKE 359.126  
**Result:** Restores normal screen after 'POKE 359.60:SCREEN 0.1'.  
**Remarks:** Not compatible with Color Basic or Disk Basic.

**Command:** POKE 359.57  
**Result:** Allows the use of graphics screen or 'SCREEN 0.1' without returning to the normal text screen.  
**Remarks:** Not compatible with Color Basic.

**Command:** POKE 359.126  
**Result:** Recovers from 'POKE 359.57'.  
**Remarks:** Not compatible with Color Basic.

**Command:** POKE 359.255  
**Result:** Disables all keys.  
**Remarks:** Any command is executed without appearing on screen. Requires ECB.

**Command:** POKE 359.126  
**Result:** Restores to normal after 'POKE 259.255'.  
**Remarks:** Requires ECB.

**Command:** POKE 359.128  
**Result:** Produces different graphic characters for different keys.  
**Remarks:** Requires ECB. Not compatible with Disk System.

**Command:** POKE 359,126

**Result:** Restores to normal after 'POKE 359,128'.

**Remarks:** This POKE will appear as graphic characters but will execute.

**Command:** POKE 359,126:POKE 360,162:POKE 361,133

**Result:** Activates double—print mode.

**Remarks:** Anything printed on the screen appears twice.

**Command:** POKE 359,57

**Result:** Returns to normal after double—print mode for Color Basic.

**Remarks:** Though this POKE appears double on screen, it will execute.

**Command:** POKE 360,&H82:POKE 361,&H73

**Result:** Returns to normal after double—print mode for ECB only.

**Remarks:** Not compatible with Disk System. See remarks for 'POKE 359,57'.

**Command:** POKE 360,&HCB:POKE 361,&H4A

**Result:** Returns to normal after double—print mode for Disk Basic 1.0.

**Remarks:** See remarks for 'POKE 359,57'. For Disk Basic 1.1 use 'POKE 360,&HCC:POKE361,&H1C'.

**Command:** POKE 359,13

**Result:** Allows the use of graphics screens and 'SCREEN 0,1' without returning to normal text screen.

**Remarks:** Requires ECB. not compatible with Disk System.

**Command:** POKE 359,126

**Result:** Returns to normal after 'POKE 359,13'.

**Remarks:** Requires ECB.

**Command:** POKE 359,57:POKE 65480,0

**Result:** Displays Page 0.

**Remarks:** Page 0 contains System variables.

**Command:** FORI = 243 TO 251:READA:POKEI,A:NEXTI:POKE416,126:  
POKE417,0:POKE418,243:DATA50,98,189,183,11,189,  
169,37,57

**Result:** 'CLS xx' Command produces different colorful designs.  
**Remarks:** xx = 1—255. Disables some ECB commands as GET/PUT,  
RENUM,etc.

**Command:** PEEK(425)

**Result:** Returns the number of characters input in the most  
recently executed string INPUT/LINEINPUT command.

**Remarks:** Not applicable to variable INPUT commands such as  
'INPUT A'.

**Command:** PEEK(465)

**Result:** Returns the no. of characters in the file to be CLOAD(M).  
**Remarks:** For ex. 'CLOADM"TITLE"' will return 5 with the above  
command.

**Command:** FORI = 466 TO 473:PRINT CHR\$(PEEK(I));:NEXTI

**Result:** Returns the filename of the file to be loaded by  
CLOAD(M).

**Remarks:** For ex. 'CLOAD"TITLE"' will return 'TITLE'.

**Command:** FORI = 474 TO 481:PRINT CHR\$(PEEK(I));:NEXTI

**Result:** Returns the filename of the most recently loaded file.

**Remarks:** None.

**Command:** A = PEEK(482):B = PEEK(483):C = PEEK(484)

**Result:** If A,B,C = 0 then the most recently 'CLOADED' File is  
Basic.

**Remarks:** The 5 filetypes are: BASIC, ASCII, DATA, ML, ML BLOCK  
LOAD.

**Command:** A = PEEK(482):B = PEEK(483):C = PEEK(484)

**Result:** If A = 0,B = 255 & C = 255, the most recent 'CLOADED' file  
is ASCII.

**Remarks:** None.

**Command:** A = PEEK(482):B = PEEK(483):C = PEEK(484)

**Result:** If A = 1,B = 255 & C = 255, the most recent 'CLOADED' file  
is a DATA File.

**Remarks:** None.

**Command:** A = PEEK(482):B = PEEK(483):C = PEEK(484)  
**Result:** If A = 2,B = 0 & C = 0, then the most recently  
'CLOADED' file is ML.  
**Remarks:** None.

**Command:** A = PEEK(482): B = PEEK(483):C = PEEK(484)  
**Result:** If A = 2,B = 0 & C = 255, then the most recently  
'CLOADED' file is Machine Language Block Load.  
**Remarks:** None.

**Command:** PEEK(485)\*256 + PEEK(486)  
**Result:** Returns the EXEC Address of a cassette-loaded  
ML Program.  
**Remarks:** None.

**Command:** PEEK(487)\*256 + PEEK(488)  
**Result:** Returns the start address of a cassette-loaded  
ML Program.  
**Remarks:** None.

**Command:** POKE (1024-1535).X  
**Result:** Stores character 'X' to Text Screen.  
**Remarks:** X = 1 to 255.

**Command:** POKE (1536-3071).X  
**Result:** Stores 'X' to location (1536-3071) in graphics  
page 1.  
**Remarks:** X = 1 to 255.

**Command:** POKE (3072-4607).X  
**Result:** Stores 'X' to location (3072-4607) in graphics  
page 2.  
**Remarks:** X = 1 to 255.

**Command:** POKE (4608-6143).X  
**Result:** Stores 'X' to location (4608-6143) in graphics  
page 3.  
**Remarks:** X = 1 to 255.

**Command:** POKE (6144—7679).X  
**Result:** Stores 'X' to location (6144—7679) in graphics page 4.  
**Remarks:** X = 1—255. For ECB only.

**Command:** POKE (7680—9215).X  
**Result:** Stores 'X' to location (7680—9215) in graphics page 5.  
**Remarks:** X = 1 to 255. For ECB only.

**Command:** POKE (9216—10751).X  
**Result:** Stores 'X' to location (9216—10751) in graphics page 6.  
**Remarks:** X = 1 to 255. For ECB only.

**Command:** POKE (10752—12287).X  
**Result:** Stores 'X' to location (10752—12287) in graphics page 7.  
**Remarks:** X = 1 to 255. For ECB only.

**Command:** POKE (12288—13823).X  
**Result:** Stores 'X' to location (12288—13823) in graphics page 8.  
**Remarks:** X = 1 to 255. For ECB only.

**Command:** EXEC 32768  
**Result:** Displays ECB copyright message.  
**Remarks:** Requires ECB. Gives disk copyright message in Disk Basic.

**Command:** PEEK (32768)  
**Result:** Returns 69 if ECB is present.  
**Remarks:** Helps to distinguish ECB from Color Basic.

**Command:** EXEC 32944  
**Result:** Displays ECB copyright message even if Disk Basic present.  
**Remarks:** Requires ECB.

**Command:** FOR I = 33021 to 33023: PRINT CHR\$(PEEK(I));:NEXT I  
**Result:** Displays ECB Version.  
**Remarks:** Requires ECB.

**Command:** FOR I = 33039 TO 33042:PRINT CHR\$(PEEK(I));:NEXT I  
**Result:** Displays ECB copyright year.  
**Remarks:** Requires ECB.

**Command:** EXEC 34471  
**Result:** Turns on Trace Flag.  
**Remarks:** Requires ECB. Same as 'TRON' Command.

**Command:** EXEC 34472  
**Result:** Turns off Trace Flag.  
**Remarks:** Requires ECB. Same as 'TROFF' Command.

**Command:** EXEC 35337  
**Result:** Renumbers any Basic pro. in memory with an increment of 10.  
**Remarks:** Requires ECB.

**Command:** EXEC 40999  
**Result:** Produces a warm start.  
**Remarks:** POKE 113,0:EXEC 40999 produces an immediate cold start.

**Command:** EXEC 41175  
**Result:** Displays Color Basic copyright message.  
**Remarks:** Can be used to determine the Color Basic version.

**Command:** FOR I = 41299 TO 41301:PRINT CHR\$(PEEK(I));:NEXT I  
**Result:** Displays the Color Basic version.  
**Remarks:** None.

**Command:** EXEC 41393

**Result:** Blinks cursor and waits for a keystroke.

**Remarks:** None.

**Command:** EXEC 42060

**Result:** Saves a basic program in memory to cassette.

**Remarks:** Same as 'CSAVE' command

**Command:** EXEC 42136

**Result:** Loads a basic program from cassette.

**Remarks:** Same as 'CLOAD' command

**Command:** EXEC 42238

**Result:** Loads a ML Program from cassette.

**Remarks:** Same as 'CLOADM' command

**Command:** EXEC 42521

**Result:** Generates an '?IO ERROR' Message.

**Remarks:** None.

**Command:** EXEC 42753

**Result:** Reads a block from cassette.

**Remarks:** None.

**Command:** EXEC 42954

**Result:** Turns on tape motor.

**Remarks:** Same as 'MOTOR ON' command

**Command:** EXEC 42961

**Result:** Generates a 6/10th of a second delay.

**Remarks:** Can be used within your Basic program as a delay statement.

**Command:** EXEC 42963

**Result:** Generates a 4/10th of a second delay.

**Remarks:** Can be used within your Basic program as a delay statement.

**Command:** EXEC 42985

**Result:** Turns off tape motor

**Remarks:** Same as 'MOTOR OFF' command.

**Command:** EXEC 43304

**Result:** Clears text screen and homes cursor.

**Remarks:** None.

**Command:** EXEC 43319

**Result:** Clears text screen and prints 'MICROSOFT'.

**Remarks:** None.

**Command:** EXEC 43380

**Result:** Turns off Audio.

**Remarks:** Same as 'AUDIO OFF' command.

**Command:** EXEC 43421

**Result:** Turns on Audio.

**Remarks:** Same as 'AUDIO ON' command

**Command:** EXEC 43486:PRINT PEEK (346)

**Result:** Returns right joystick horizontal value.

**Remarks:** Joystick values are between 0 and 63.

**Command:** EXEC 43486:PRINT PEEK (347)

**Result:** Returns right joystick vertical value.

**Remarks:** Joystick values are between 0 and 63.

**Command:** EXEC 43486:PRINT PEEK(348)  
**Result:** Returns left joystick horizontal value.  
**Remarks:** Joystick values are between 0 & 63.

**Command:** EXEC 43486:PRINT PEEK(349)  
**Result:** Returns left joystick vertical value.  
**Remarks:** Joystick values are between 0 & 63.

**Command:** EXEC 44313  
**Result:** Erases any Basic program in memory.  
**Remarks:** Same as 'NEW' command.

**Command:** EXEC 44321  
**Result:** Starts the execution of a Basic program in memory.  
**Remarks:** Same as 'RUN' command.

**Command:** EXEC 44326  
**Result:** Erases all variables and strings.  
**Remarks:** None.

**Command:** EXEC 44539  
**Result:** Waits for a keystroke.  
**Remarks:** 'EXEC 44539' is used as a substitute for the following: 10 A\$ = INKEY\$: IF A\$ = "" Then 10

**Command:** EXEC 44539:PRINT CHR\$(PEEK(135))  
**Result:** Waits for keystroke and returns the key pressed.  
**Remarks:** None.

**Command:** EXEC 44658  
**Result:** Generates an '?OM ERROR' Message.  
**Remarks:** None.

**Command:** EXEC 44661

**Result:** 'RUN's a basic program currently in memory.

**Remarks:** Same as 'RUN' command.

**Command:** EXEC 44759

**Result:** Generates an '?SN ERROR' Message.

**Remarks:** None.

**Command:** EXEC 45045 AS

**Result:** Waits for an input and stores the data input in AS.

**Remarks:** Cannot be used from Direct Mode. Same as 'INPUT AS' command.

**Command:** EXEC 46942

**Result:** 'LIST's a Basic program in memory to printer.

**Remarks:** Same as 'LLIST' command.

**Command:** EXEC 46946

**Result:** 'LIST's a Basic program in memory to screen.

**Remarks:** Same as 'LIST' command.

**Command:** EXEC 47532

**Result:** Prints a space.

**Remarks:** None.

**Command:** PEEK(65280)

**Result:** Returns 255 or 127 if no fire buttons are pressed.

**Remarks:** None.

**Command:** PEEK(65280)

**Result:** Returns 125 or 253 if left joystick fire button is pressed.

**Remarks:** Requires joystick(s).

**Command:** PEEK(65280)

**Result:** Returns 126 or 254 if right joystick fire button is pressed.

**Remarks:** Requires joystick(s).

**Command:** PEEK(65280)

**Result:** Returns 124 or 252 if both joystick fire buttons are pressed.

**Remarks:** Requires 2(two) joysticks.

**Command:** PEEK(65282)

**Result:** Returns 255 if a key is pressed or 0 if no key is pressed.

**Remarks:** Can be used as a substitute for the 'INKEY\$' command.

**Command:** POKE 65313. (PEEK(65313) OR 8)

**Result:** Turns on cassette motor.

**Remarks:** Same as 'MOTOR ON' command.

**Command:** POKE 65313.3

**Result:** Turns on motor relay.

**Remarks:** None.

**Command:** POKE 65313.60

**Result:** Turns on motor relay.

**Remarks:** None.

**Command:** POKE 65313.52

**Result:** Turns off motor relay.

**Remarks:** None.

**Command:** FOR I = 1 TO 30:POKE 65313.3:POKE 65313.52:NEXT I

**Result:** Produces a 'CRICKET SOUND' in Coco1.

**Remarks:** This command does not work if printer is on.

**Command:** (PEEK(65314) AND 1)

**Result:** Returns 1 if printer is not ready.

**Remarks:** Can be used to determine if the printer is on or off.

**Command:** PEEK (65314)

**Result:** Returns an even number if printer is on.

**Remarks:** Can be used to determine if the printer is on or off.

**Command:** POKE 65315.54

**Result:** Disables auto execute from ROMPAK.

**Remarks:** \* Danger \* Plugging in rompak with power on could damage the computer and/or rompak.

**Command:** POKE 65315.55

**Result:** Enables auto execute from ROMPAK after 'POKE 65315.54'.

**Remarks:** Same as 'POKE 65315.54'.

**Command:** POKE 65495.0

**Result:** Hi-Speed poke. Doubles the speed of all commands/functions.

**Remarks:** \* Danger \* This poke heats up the computer chips and rarely works with disk. Cannot be used during cassette I/O.

**Command:** POKE 65494.0

**Result:** Restores to normal after 'POKE 65495.0'.

**Remarks:** None.

**Command:** POKE 65497.0

**Result:** Hi-Speed poke. Loss of screen/faster than 'POKE 65495.0'.

**Remarks:** Same as for 'POKE 65495.0'.

**Command:** POKE 65496.0

**Result:** Restores to normal after 'POKE 65497.0'.

**Remarks:** This poke won't appear on the screen, but will execute.

## GRAPHICS/SEMIGRAPHIC MODES

**Command:** POKE 65314, (PEEK(65314) AND 7):POKE 65479,0:POKE 65481,0:POKE 65483,0:POKE 65476,0:POKE 65474,0:POKE 65472,0

**Result:** Sets semigraphic 4 mode (32 x 16 —— 8 colors).  
**Remarks:** 'POKE' in locations 3584 thru 4096 to set pixel(s).

**Command:** POKE 65314,(PEEK(65314) AND 7) + 8:POKE 65479,0:POKE 65481,0:POKE 65483,0:POKE 65476,0:POKE 65474,0:POKE 65472,0

**Result:** Sets semigraphic 6 mode (32 x 16 —— CSS 0, 4 colors).

**Remarks:** 'POKE' in locations 3584 thru 4096 to set pixels.

**Command:** POKE 65314, (PEEK(65314) AND 7) + 24: POKE 65479,0:POKE 65481,0:POKE 65483,0:POKE 65476,0:POKE 65474,0:POKE 65472,0

**Result:** Sets semigraphic 6 mode (32 x 16 —— CSS 1,4 colors).

**Remarks:** 'POKE' in locations 3584 thru 4096 to set pixels.

**Command:** POKE 65314, (PEEK(65314) AND 7):POKE 65479,0:POKE 65481,0:POKE 65483,0:POKE 65476,0:POKE 65475,0:POKE 65472,0

**Result:** Sets semigraphic 8 mode (64 x 64 —— 8 colors).  
**Remarks:** 'POKE' in locations 3584 thru 5632 to set pixels.

**Command:** POKE 65314,(PEEK(65314) AND 7):POKE 65479,0:POKE 65481,0:POKE 65483,0:POKE 65477,0:POKE 65474,0:POKE 65472,0

**Result:** Sets semigraphic 12 mode (64 x 96 —— 8 colors).  
**Remarks:** 'POKE' in locations 3584 thru 6656 to set pixels.

**Command:** POKE 65314, (PEEK(65314) AND 7):POKE 65479,0:POKE 65481,0:POKE 65483,0:POKE 65477,0:POKE 65475,0:POKE 65472,0

**Result:** Sets semigraphic 24 mode (64 x 192 —— 8 colors).  
**Remarks:** 'POKE' in locations 3584 thru 9728 to set pixels.

**Command:** POKE 65314, (PEEK(65314) AND 7) + 128:POKE  
65479,0:POKE 65481,0:POKE 65483,0:POKE  
65473,0:POKE 65474,0:POKE 65476,0  
**Result:** Sets 64 x 64 —— 4 color graphics mode with CSS 0.  
**Remarks:** 'POKE' in locations 3584 thru 4608 to set pixels.

**Command:** POKE 65314, (PEEK(65314) AND 7) + 136:POKE  
65479,0:POKE 65481,0:POKE 65483,0:POKE  
65473,0:POKE 65474,0:POKE 65476,0  
**Result:** Sets 64 x 64 —— 4 color graphics mode with CSS 1.  
**Remarks:** 'POKE' in locations 3584 thru 4608 to set pixels.

**Command:** POKE 65314, (PEEK(65314) AND 7) + 144:POKE  
65479,0:POKE 65481,0:POKE 65483,0:POKE  
65473,0:POKE 65474,0:POKE 65476,0  
**Result:** Sets 128 x 64 —— 2 color graphics mode with CSS 0.  
**Remarks:** 'POKE' in locations 3584 thru 4608 to set pixels.

**Command:** POKE 65314, (PEEK(65314) AND 7) + 152:POKE  
65479,0:POKE 65481,0:POKE 65483,0:POKE  
65473,0:POKE 65474,0:POKE 65476,0  
**Result:** Sets 128 x 64 —— 2 color graphics mode with CSS 1.  
**Remarks:** 'POKE' in locations 3584 thru 4608 to set pixels.

**Command:** POKE 65314, (PEEK(65314) AND 7) + 160:POKE  
65479,0:POKE 65481,0:POKE 65483,0:POKE  
65472,0:POKE 65475,0:POKE 65476,0  
**Result:** Sets 128 x 64 —— 4 color graphics mode with CSS 0.  
**Remarks:** 'POKE' in locations 3584 thru 4608 to set pixels.

**Command:** POKE 65314, (PEEK(65314) AND 7) + 168:POKE  
65479,0:POKE 65481,0:POKE 65483,0:POKE  
65472,0:POKE 65475,0:POKE 65476,0  
**Result:** Sets 128 x 64 —— 4 color graphics mode with CSS 1.  
**Remarks:** 'POKE' in locations 3584 thru 4608 to set pixels.

**Command:** POKE 65314, (PEEK(65314) AND 7) + 176:POKE  
65479,0:POKE 65481,0:POKE 65483,0:POKE  
65476,0:POKE 65475,0:POKE 65473,0  
**Result:** Sets 128 x 96 —— 2 color graphics mode with CSS 0.  
**Remarks:** 'POKE' in locations 3584 thru 5120 to set pixels.

**Command:** POKE 65314, (PEEK(65314) AND 7) + 184:POKE  
65479,0:POKE 65481,0:POKE 65483,0:POKE  
65476,0:POKE 65475,0:POKE 65473,0  
**Result:** Sets 128 x 96 —— 2 color graphics mode with CSS 1.  
**Remarks:** 'POKE' in locations 3584 thru 5120 to set pixel(s).

**Command:** POKE 65314,(PEEK(65314) AND 7) + 160:POKE  
65479,0:POKE 65481,0:POKE 65483,0:POKE  
65472,0:POKE 65475,0:POKE 65476,0  
**Result:** Sets 128 x 96 —— 4 color graphics mode with CSS 0.  
**Remarks:** 'POKE' in locations 3584 thru 5632 to set pixels.

**Command:** POKE 65314, (PEEK(65314) AND 7) + 168: POKE  
65479,0:POKE 65481,0:POKE 65483,0:POKE  
65472,0:POKE 65475,0:POKE 65476,0  
**Result:** Sets 128 x 96 —— 4 color graphics mode with CSS 1.  
**Remarks:** 'POKE' in locations 3584 thru 5632 to set pixels.

**Command:** POKE 65314, (PEEK(65314) AND 7) + 192:POKE  
65479,0:POKE 65481,0:POKE 65483,0:POKE  
65477,0:POKE 65474,0:POKE 65472,0  
**Result:** Sets 128 x 96 —— 4 color graphics mode with CSS 0.  
**Remarks:** 'POKE' in locations 3584 thru 6656 to set pixels.

**Command:** POKE 65314,(PEEK(65314) AND 7) + 200:POKE  
65479,0:POKE 65481,0: POKE 65483,0:POKE  
65477,0:POKE 65474,0:POKE 65472,0  
**Result:** Sets 128 x 96 —— 4 color graphics mode with CSS 1.  
**Remarks:** 'POKE' in locations 3584 thru 6656 to set pixels.

**Command:** POKE 65314, (PEEK(65314) AND 7) + 208:POKE  
65479,0:POKE 65481,0: POKE 65483,0:POKE  
65477,0:POKE 65474,0:POKE 65473,0  
**Result:** Sets 128 x 192 —— 2 color graphics mode with  
CSS 0.  
**Remarks:** 'POKE' in locations 3584 thru 6656 to set pixels.

**Command:** POKE 65314,(PEEK(65314)AND7) + 216:POKE  
65479,0:POKE 65481,0:POKE 65483,0:POKE  
65477,0:POKE 65474,0:POKE 65473,0  
**Result:** Sets 128 x 192 —— 2 color graphics mode with  
CSS 1.  
**Remarks:** 'POKE' in locations 3584 thru 6656 to set pixels.

**Command:** POKE 65314, (PEEK(65314) AND 7) + 224:POKE 65479,0:POKE 65481,0:POKE 65483,0:POKE 65477,0:POKE 65475,0:POKE 65472,0

**Result:** Sets 128 x 192 —— 4 color graphics mode with CSS 0.

**Remarks:** 'POKE' in locations 3584 thru 9728 to set pixels.

**Command:** POKE 65314, (PEEK(65314) AND 7) + 232:POKE 65479,0:POKE 65481,0:POKE 65483,0:POKE 65477,0:POKE 65475,0:POKE 65472,0

**Result:** Sets 128 x 192 —— 4 color graphics mode with CSS 1.

**Remarks:** 'POKE' in locations 3584 thru 9728 to set pixels.

**Command:** POKE 65314, (PEEK(65314) AND 7) + 240:POKE 65479,0:POKE 65481,0:POKE 65483,0:POKE 65477,0:POKE 65475,0:POKE 65472,0

**Result:** Sets 256 x 192 —— 2 color graphics mode with CSS 0.

**Remarks:** 'POKE' in locations 3584 thru 9728 to set pixels.

**Command:** POKE 65314,(PEEK(65314) AND 7) + 248:POKE 65479,0:POKE 65481,0:POKE 65483,0:POKE 65477,0:POKE 65475,0:POKE 65472,0

**Result:** Sets 256 x 192 —— 2 color graphics mode with CSS 1.

**Remarks:** 'POKE' in locations 3584 thru 9728 to set pixels.

**Command:** PMODE 3,1:SCREEN 1,0:POKE 65314,240

**Result:** Lets you use 4 colors from PMODE 4. Requires ECB.

**Remarks:** 4 colors are Drk. Green, Medium Green, Gray Green & Light Green.

**Command:** PMODE 3,1:SCREEN 1,1:POKE 65314,248

**Result:** Lets you use 4 colors from PMODE 4. Requires ECB.

**Remarks:** 4 colors are Black, Red, Blue and White.

**Command:** POKE 65314,200

**Result:** Returns to PMODE 3,1:SCREEN 1,1 after the preceding POKE.

**Remarks:** Requires ECB.

**Command:** POKE 65314.192

**Result:** Returns to PMODE 3 after 'PMODE3:SCREEN 1.0:POKE 65314.240'.

**Remarks:** Requires ECB.

**Command:** PMODE 1.1:SCREEN 1.0:POKE 65314.240

**Result:** Lets you use 4 colors from PMODE 1. Requires ECB.

**Remarks:** 4 colors are Drk. Green. Medium Green. Gray Green & Light Green.

**Command:** PMODE 1.1:SCREEN 1.1:POKE 65314.248

**Result:** Lets you use 4 colors from PMODE 1. Requires ECB.

**Remarks:** 4 colors are Black, Red, Blue and White.

**Command:** POKE 65314.192

**Result:** Returns to PMODE 1 after 'PMODE 1.1:SCREEN1, 0:POKE 65314.240'.

**Remarks:** Requires ECB.

**Command:** POKE 65314.200

**Result:** Returns to PMODE 1 after 'PMODE1.1:SCREEN1, 1:POKE 65314.248'.

**Remarks:** Requires ECB.

**Note:** For detailed description of the GRAPHIC/SEMIGRAPHIC MODES, you may consult pages 264 to 266 of 'GETTING STARTED WITH COLOR BASIC'.

#### **FOR 4 COLOR GRAPHIC MODES:**

CSS 0 = Green, Yellow, Blue and Red.

CSS 1 = Buff, Cyan, Magneta and Orange.

#### **FOR 2 COLOR GRAPHIC MODES:**

CSS 0 = Black and Green.

CSS 1 = Black and Buff.

## UTILITY ROUTINES

**Command:** POKE 288,53:FOR I=3584 TO 3605:READ A:POKE I,A:NEXT I:EXEC 3584:CLEAR 200,PEEK(116)\*256 + PEEK(117)-449:DATA 142,170,102,222,116, 51,201,254,64,255,1,33,166,128,167,192, 140,171,26,37,247,57

**Result:** Sets the base for Color Basic commands disables.

**Remarks:** Any one or more disables for Color Basic commands may be used following the command given above.

**Command:** POKE 32384,68

**Result:** Disables 'POKE' command for 32K/64K.

**Remarks:** Use after Color Basic base POKES. For 16K use 'POKE 16000,68'.

**Command:** PDKE 32384,79

**Result:** Restores 'POKE' command for 32K/64K.

**Remarks:** Use after Color Basic base POKES. For 16K use 'POKE 16000,79'.

**Command:** POKE 32392,68

**Result:** Disables 'LIST' command for 32K/64K.

**Remarks:** Use after Color Basic base POKES. For 16K use 'POKE 16008,68'.

**Command:** POKE 32392,73

**Result:** Restores 'LIST' command for 32K/64K.

**Remarks:** Use after Color Basic base POKES. For 16K use 'POKE 16008,73'.

**Command:** POKE 32401,68

**Result:** Disables 'NEW' command for 32K/64K.

**Remarks:** Use after Color Basic base POKES. For 16K use 'POKE 16017,68'.

**Command:** POKE 32401.69  
**Result:** Restores 'NEW' command for 32K/64K.  
**Remarks:** Use after Color Basic base POKES. For 16K use 'POKE 16017.69'.

**Command:** POKE 32404.68  
**Result:** Disables 'CLOAD(M)' command for 32K/64K.  
**Remarks:** Use after Color Basic base POKES. For 16K use 'POKE 16020.68'

**Command:** POKE 32404.76  
**Result:** Restores 'CLOAD(M)' command for 32K/64K.  
**Remarks:** Use after Color Basic base POKES. For 16K use 'POKE 16020.79'.

**Command:** POKE 32409.68  
**Result:** Disables 'CSAVE(M)' command for 32K/64K.  
**Remarks:** Use after Color Basic base POKES. For 16K use 'POKE 16025.68'.

**Command:** POKE 32409.83  
**Result:** Restores 'CSAVE(M)' command for 32K/64K.  
**Remarks:** Use after Color Basic base POKES. For 16K use 'POKE 16025.83'.

**Command:** POKE 32423.68  
**Result:** Disables 'LLIST' command for 32K/64K.  
**Remarks:** Use after Color Basic base POKES. For 16K use 'POKE 16039.68'.

**Command:** POKE 32423.76  
**Result:** Restores 'LLIST' command for 32K/64K.  
**Remarks:** Use after Color Basic base POKES. For 16K use 'POKE 16039.76'.

**Command:** POKE 32454.68  
**Result:** Disables 'EXEC' command for 32K/64K.  
**Remarks:** Use after Color Basic base POKES. For 16K use 'POKE 16070.68'.

**Command:** POKE 32454,88

**Result:** Restores 'EXEC' command for 32K/64K.

**Remarks:** Use after Color Basic base POKES. For 16K use 'POKE 16070,88'.

**Note:** Once you have typed in the base pokes for Color Basic Commands disables, you may use any ONE or SEVERAL Command Disable POKES.

**Command:** FOR I = 3584 TO 3605:READ A:POKE I,A:NEXT I: EXEC 3584: CLEAR 200,PEEK(116)\*256 + PEEK(117)-561:DATA 142,129,131,222,116,51,201,253,208,255,1,43,166,128,167,192,140,129,240,37,247,57

**Result:** Sets the base for ECB Commands Disables.  
Requires ECB.

**Remarks:** Any one or more of the ECB Command Disables may be used following the command given above.

**Command:** POKE 32207,68

**Result:** Disables 'DEL' command for 32K/64K.

**Remarks:** Requires ECB. Use after ECB base POKES. For 16K use 'POKE 15823,68'.

**Command:** POKE 32207,69

**Result:** Restores 'DEL' command for 32K/64K. Requires ECB.

**Remarks:** Use after ECB base POKES. For 16K use 'POKE 15823,69'.

**Command:** POKE 32211,68

**Result:** Disables 'EDIT' command for 32K/64K. Requires ECB.

**Remarks:** Use after ECB base POKES. For 16K use 'POKE 15827,68'.

**Command:** POKE 32211,73

**Result:** Restores 'EDIT' command for 32K/64K. Requires ECB.

**Remarks:** Use after ECB base POKES. For 16K use 'POKE 15827,73'.

**Command:** POKE 32214.68  
**Result:** Disables 'TRON' command for 32K/64K.  
Requires ECB.  
**Remarks:** Use after ECB base POKES. For 16K use 'POKE 15830.68'.

**Command:** POKE 32214.82  
**Result:** Restores 'TRON' command for 32K/64K.  
Requires ECB.  
**Remarks:** Use after ECB base POKES. For 16K use 'POKE 15830.82'.

**Command:** POKE 32218.68  
**Result:** Disables 'TROFF' command for 32K/64K.  
Requires ECB.  
**Remarks:** Use after ECB base POKES. For 16K use 'POKE 15834.68'.

**Command:** POKE 32218.82  
**Result:** Restores 'TROFF' command for 32K/64K.  
Requires ECB.  
**Remarks:** Use after ECB base POKES. For 16K use 'POKE 15834.82'.

**Command:** POKE 32253.68  
**Result:** Disables 'PCLEAR' command for 32K/64K.  
Requires ECB.  
**Remarks:** Use after ECB base POKES. For 16K use 'POKE 15869.68'.

**Command:** POKE 32253.67  
Restores 'PCLEAR' command for 32K/64K.  
Requires ECB.  
**Remarks:** Use after ECB base POKES. For 16K use 'POKE 15869.67'.

**Command:** POKE 32299.68  
**Result:** Disables 'DLOAD' command for 32K/64K. Requires ECB.  
**Remarks:** Use after ECB base POKES. For 16K use 'POKE 15915.68'.

**Command:** POKE 32299.76  
**Result:** Restores 'DLOAD' command for 32K/64K.  
Requires ECB.  
**Remarks:** Use after ECB base POKES. For 16K use 'POKE 15915.76'.

**Command:** POKE 32304,68

Disables 'RENUM' command for 32K/64K.

Requires ECB.

**Result:** Use after ECB base POKES. For 16K use 'POKE

**Remarks:** 15920,68'.

**Command:** POKE 32304,69

Restores 'RENUM' command for 32K/64K.

Requires ECB.

**Result:** Use after ECB base POKES. For 16K use 'POKE

**Remarks:** 15920,69'.

**Command:** POKE 32311,68

Disables 'PRINT USING' command for 32K/64K.

Requires ECB.

**Result:** Use after ECB base POKES. For 16K use 'POKE

**Remarks:** 15927,68'.

**Command:** POKE 32311,83

**Result:** Restores 'PRINT USING' command for 32K/64K.

Requires ECB.

**Remarks:** Use after ECB base POKES. For 16K use 'POKE

15927,83'.

**Note:** Once you have typed in the base POKES for ECB Commands Disables, you may use any ONE or SEVERAL ECB Command Disable POKES.

**Command:** For I=243 TO 250: READ A:POKE I,A:NEXT I:POKE 359, 126:POKE 360,0:POKE 361,243:DATA 52,20,189,162,191,53,20,57

**Result:** Enters Screen-To-Printer Mode.

**Remarks:** Everything that is printed on the screen goes to printer.

**Command:** POKE 359,57

**Result:** Returns to normal after Screen-To-Printer Mode.

**Remarks:** For Color Basic only.

**Command:** POKE 360,&H82:POKE 361,&H73

**Result:** Returns to normal after Screen-To-Printer Mode.

**Remarks:** For ECB only.

**Command:** POKE 360,&HCB:POKE 361,&H4A

**Result:** Returns to normal after Screen-To-Printer Mode.

**Remarks:** For Disk Basic only. For Disk Basic 1.1 use 'POKE 360,&HCC:POKE 361,&H1C'

**Command:** A = PEEK(116)\*256 + PEEK(117)-600:X = INT(A/256):  
Y = A-(X\*256):FOR I = A TO A + 37:READ B:POKE I,B:  
NEXT I:DATA 50,100,189,161,121,132,127,129,10,  
39,11,129,9,39,18,129,12,38,13,126,160,39,182,  
255,33,138,8,183,255,33,134,10,57,189,167,233,32,  
248:POKE 362,126:POKE 363,X:POKE 364,Y:CLEAR 200,A

**Result:** Allows RESET, MOTOR ON and MOTOR OFF from the keyboard.

**Remarks:** For Color Basic only. Use CLEAR-KEY for RESET, down arrow for MOTOR ON, and right arrow for MOTOR OFF. This command being lengthy can be broken into 2 statements after the data.

**Command:** A = PEEK(116)\*256 + PEEK(117)-602:X = INT(A/256):  
Y = A-(X\*256):FOR I = A TO A + 40:READ B:POKE I,B:  
NEXT I:DATA 50,100,189,140,241,189,161,121,132,  
127,129,10,39,11,129,9,39,18,129,12,38,13,126,  
160,39,182,255,33,138,8,183,255,33,134,10,57,  
189,167,233,32,248:POKE 363,X:POKE 364,Y: CLEAR 200,A

**Result:** Allows reset, MOTOR ON and MOTOR OFF from the keyboard.

**Remarks:** For ECB only. Also see remarks for the preceding command.

**Command:** POKE 362,57

**Result:** Recovers from Color Basic POKES for RESET, MOTOR ON & MOTOR OFF.

**Remarks:** For Color Basic only.

**Command:** POKE 363,&H8C:POKE 364,&HF1

**Result:** Recovers from ECB POKES for RESET, MOTOR ON and MOTOR OFF.

**Remarks:** For ECB only.

**Command:** POKE 363,&HC5:POKE 364,&H8F

**Result:** Recovers from ECB POKES for reset, motor on and motor off.

**Remarks:** For Disk Basic only. For Disk Basic 1.1, use 'POKE 363,&HC5:POKE 364,&HBC'.

**Command:**  $A = \text{PEEK}(116) * 256 + \text{PEEK}(117) - 621$ :  $X = \text{INT}(A/256)$ :  
 $Y = A - (X * 256)$ : FOR  $I = A$  TO  $A + 16$ : READ  $B$ : POKE  $I, B$ :  
NEXT  $I$ : POKE 362, 126: POKE 363,  $X$ : POKE 364,  $Y$ :  
DATA 50, 100, 18, 18, 18, 189, 161, 121, 132, 127, 129,  
10, 38, 2, 134, 10, 57: CLEAR 200, A

**Result:** Sets the Base for Key Disables. For Color Basic only.

**Remarks:** Any one of the key disables may be used following the command given above.

**Command:**  $A = \text{PEEK}(116) * 256 + \text{PEEK}(117) - 621$ :  $X = \text{INT}(A/256)$ :  
 $Y = A - (X * 256)$ : FOR  $I = A$  TO  $A + 16$ : READ  $B$ : POKE  $I, B$ :  
NEXT  $I$ : POKE 363,  $X$ : POKE 364,  $Y$ : DATA 50, 100, 189, 140,  
241, 189, 161, 121, 132, 127, 129, 10, 38, 2, 134, 10, 57:  
CLEAR 200, A

**Result:** Sets the Base for Key Disables. For ECB only.

**Remarks:** Same as for preceding command.

**Command:** POKE 32156, 3

**Result:** Disables the Break Key for 32K/64K.

**Remarks:** Use after Key Disable base POKES. For 16K use 'POKE 15772, 3'.

**Command:** POKE 32156, 8

**Result:** Disables the Backspace Key for 32K/64K.

**Remarks:** Use after Key Disable Base POKES. For 16K use 'POKE 15772, 8'.

**Command:** POKE 32156, 12

**Result:** Disables the Clear Key for 32K/64K.

**Remarks:** Use after Key Disable base POKES. For 16K use 'POKE 15772, 12'.

**Command:** POKE 32156, 21

**Result:** Disables the 'SHIFT—BACKSPACE' Key for 32K/64K.

**Remarks:** Use after Key Disable base POKES. For 16K use 'POKE 15772, 21'.

**Command:** POKE 32156, 32

**Result:** Disables the Space—Bar for 32K/64K.

**Remarks:** Use after Key Disable base POKES. For 16K use 'POKE 15772, 32'.

**Command:** POKE 32156.94

**Result:** Disables the '↑' Key for 32K/64K.

**Remarks:** Use after Key Disable base POKEs. For 16K use 'POKE 15772.94'.

**Note:** Once you typed in the base pokes for the key disables, you may use any ONE of the key disable POKE commands. The key disables work ONLY with INPUT/LINEINPUT commands.

**Command:** FOR I = 330 TO 336:READ A:POKE I,A:NEXT I: DATA 50,98,28,175,126,173,165:POKE 410,126:POKE 411,1:POKE 412,74

**Result:** Disables the Break Key on all but Input/Lineinput commands.

**Remarks:** Also disables the CSAVE,TRON and TROFF commands in ECB.

**Command:** POKE 410,57

**Result:** Enables the Break Key after the Break Key Disable.

**Remarks:** For Color Basic only.

**Command:** POKE 411,&H82:POKE 412,&HB9

**Result:** Enables the Break Key after the Break Key Disable.

**Remarks:** For ECB and Disk Basic only.

**Command:** A = PEEK(116) • 256 + PEEK(117) - 632:X = INT(A/256):Y = A - (X • 256): FOR I = A TO A + 9:READ B:POKE I,B:NEXT I:POKE 362,126:POKE 363,X:POKE 364,Y:CLEAR 200,A:DATA 52,118,198,1,189,169,81,53,118,57

**Result:** Activates the Key—Beep routine.

**Remarks:** Any key pressed produces a beep. Use 'SOUND xx.xx' to create beep variations.

**Command:** POKE 362,57

**Result:** Returns to normal after Key Beep.

**Remarks:** For Color Basic only.

**Command:** POKE 363,&H8C:POKE 364,&HF1  
**Result:** Returns to normal after Key Beep for ECB only.  
**Remarks:** For Disk Basic 1.0, use 'POKE 363,&HC5:POKE 364,&H8F'. For Disk Basic 1.1, use 'POKE 363,&HC5:POKE 364,&HBC'.

**Command:** A = PEEK(116)\*256 + PEEK(117)-644:X = INT(A/256):Y = A-(X\*256):FOR I = A TO A + 9:READ B:POKE I,B:NEXT I:POKE 398,126:POKE 399,X:POKE 400,Y:CLEAR 200,A:DATA 52,118,198,1,189,169,81,53,118,57  
**Result:** Activates the Error-Beep routine.  
**Remarks:** Any error produces a beep. Use 'SOUND xx.xx' to create beep variations.

**Command:** POKE 398,57  
**Result:** Restores to normal after Error-Beep routine.  
**Remarks:** None.

**Command:** A = PEEK(116)\*256 + PEEK(117)-12:X = INT(A/256):Y = A-(X\*256):POKE 113,85:POKE 114,X:POKE 115,Y:DATA 18,189,173,33,189,172,239,126,173,158  
**Result:** Restarts your Basic Program when the Reset Button is pressed.  
**Remarks:** For Color Basic only.

**Command:** A = PEEK(116)\*256 + PEEK(117)-20:X = INT(A/256):Y = A-(X\*256):POKE 113,85:POKE 114,X:POKE 115,Y:FOR I = A TO A + 17:READ B:POKE I,B:NEXT I:DATA 18,182,255,3,138,1,183,255,3,189,173,33,189,172,239,126,173,158  
**Result:** Restarts your Basic Program when the Reset Button is pressed.  
**Remarks:** For ECB only.

**Command:** POKE 113,0  
**Result:** Recovers from the two preceding POKEs.  
**Remarks:** None.

**Command:** FOR I = 3584 TO 3612:READ A:POKE I,A:NEXTI: DATA 158,25,51,4,166,192,38,252,239,132,230,192, 38,252,230,192,38,248,230,192,38,244,223,27,223, 29,223,31,57:CSAVEM "RECOVER",3584,3613,3584

**Result:**  
**Remarks:** Recovers Basic Programs lost by 'NEW' or faulty reset. For ECB only. Insert a blank tape in your Cassette Player, press play & record and 'RUN' the above program. The 'RECOVER' routine will be saved to tape. Whenever you lose your Basic Program, 'CLOADM' the Recover Routine and type 'EXEC'.

**Command:** A = PEEK(116) \* 256 + PEEK(117) - 100:X = INT (A/256):Y = A - (X \* 256):FOR I = A TO A + 72:READ B: POKE I,B:NEXT I: POKE 362,126:POKE 363,X: POKE 364,Y:CLEAR 200,A:DATA 50,100,52,20, 189,161,121,132,127,52,2,189,161,193,142,4, 254,189,167,211,182,1,83,129,191,38,37:DATA 53,22,150,255,129,8,38,7,90,39,31,48,31,32, 15,129,32,37,22,129,123,36,18,193,249,36,14, 167,128,92,189,162,130,52,20,32,201,53,22, 151,255,57,198,1,32,251

**Result:**  
**Remarks:** Lets you repeat a key with a single keystroke. This command being lengthy can be divided into 2 statements at the beginning of the second data statement. In CoCo1, use press 'CLEAR' to start Key Repeat and release it to stop the Key Repeat. In CoCo2, use 'CLEAR' to start Key Repeat and any other key to stop the Key Repeat.

**Command:** FOR I = 3584 TO 3642:READ A:POKE I,A:NEXTI:DATA 12,104,127,1,209,189,166,76,158,25,48,31,111, 128,189,173,25,189,167,124,158,25,159,126,220, 126:DATA 76,189,172,55,189,167,11,16,38,150, 211,150,124,16,39,150,205,42,233,159,27,189, 167,233,189,173,33,189,172,239,126,173, 158:CSAVEM "AUTOLOAD",3584,3643,3584

**Result:**  
**Remarks:** Autostarts your basic program. For ECB only. To use: Insert a blank cassette in Cassette Player, press play & record and 'RUN' the above program. The 'AUTOLOAD' routines will be saved to your cassette. Now LOAD or CLOAD your Basic Program and CSAVE it following the 'AUTOLOAD' routine. Now rewind the cassette on which the 'AUTOLOAD' routine & your Basic program is 'CSAVED', press play and type: CLOADM: EXEC. Your Basic Program will 'RUN' automatically.

## PRINTER

**Command:** POKE 149,4:POKE 150,88

**Result:** Sets baud rate 50 for printer.

**Remarks:** Consult the printer manual for your printer baud rates.

**Command:** POKE 149,2:POKE 150,227

**Result:** Sets baud rate 75 for printer.

**Remarks:** Consult the printer manual for your printer baud rates.

**Command:** POKE 149,1:POKE 150,246

**Result:** Sets baud rate 110 for printer.

**Remarks:** Consult the printer manual for your printer baud rates.

**Command:** POKE 149,1:POKE 150,153

**Result:** Sets baud rate 134.5 for printer.

**Remarks:** Consult the printer manual for your printer baud rates.

**Command:** POKE 149,1:POKE 150,110

**Result:** Sets baud rate 150 for printer.

**Remarks:** Consult the printer manual for your printer baud rates.

**Command:** POKE 149,0:POKE 150,180

**Result:** Sets baud rate 300 for printer.

**Remarks:** Consult the printer manual for your printer baud rates.

**Command:** POKE 149,0:POKE 150,87

**Result:** Sets baud rate 600 for printer.

**Remarks:** Consult the printer manual for your printer baud rates.

**Command:** POKE 149,0:POKE 150,40  
**Result:** Sets baud rate 1200 for printer.  
**Remarks:** Consult the printer manual for your printer baud rates.

**Command:** POKE 149,0:POKE 150,25  
**Result:** Sets baud rate 1800 for printer.  
**Remarks:** Consult the printer manual for your printer baud rates.

**Command:** POKE 149,0:POKE 150,23  
**Result:** Sets baud rate 2000 for printer.  
**Remarks:** Consult the printer manual for your printer baud rates.

**Command:** POKE 149,0:POKE 150,18  
**Result:** Sets baud rate 2400 for printer.  
**Remarks:** Consult the printer manual for your printer baud rates.

**Command:** POKE 149,0:POKE 150,10  
**Result:** Sets baud rate 3600 for printer.  
**Remarks:** Consult the printer manual for your printer baud rates.

**Command:** POKE 149,0:POKE 150,7  
**Result:** Sets baud rate 4800 for printer.  
**Remarks:** Consult the printer manual for your printer baud rates.

**Command:** POKE 149,0:POKE 150,3  
**Result:** Sets baud rate 7200 for printer.  
**Remarks:** Consult the printer manual for your printer baud rates.

**Command:** POKE 149,0:POKE 150,1  
**Result:** Sets baud rate 9600 for printer.  
**Remarks:** Consult the printer manual for your printer baud rates.

**Command:** POKE 151,0:POKE 152,1

**Result:** Sets 0 Second line return delay for printer.

**Remarks:** None.

**Command:** POKE 151,64:POKE 152,0

**Result:** Sets .288 second line return delay for printer.

**Remarks:** None.

**Command:** POKE 151,128:POKE 152,0

**Result:** Sets .576 second line return delay for printer.

**Remarks:** None.

**Command:** POKE 151,255:POKE 152,255

**Result:** Sets 1.15 seconds line return delay for printer.

**Remarks:** None.

**Command:** POKE 155,16

**Result:** Sets printer line width to 16 characters per line.

**Remarks:** None.

**Command:** POKE 155,32

**Result:** Sets printer line width to 32 characters per line.

**Remarks:** None.

**Command:** POKE 155,64

**Result:** Sets printer line width to 64 characters per line.

**Remarks:** None.

**Command:** POKE 155,132

**Result:** Sets printer line width to 132 characters per line.

**Remarks:** 132 is basic default printer line width.

**Command:** POKE 155,255

**Result:** Sets printer line width to 255 characters per line.

**Remarks:** None

**Command:** POKE 111,254:PRINT "THIS IS A TEST"

**Result:** Prints the message 'THIS IS A TEST' to printer.

**Remarks:** 254 is the device number for printer.

**Command:** POKE 111,254:DIR

**Result:** Prints the disk directory to printer.

**Remarks:** For Disk System only.

**Command:** POKE 111,254:EXEC 46946

**Result:** 'LIST's a Basic program in memory to printer.

**Remarks:** Same as 'LLIST' Command

**Command:** PEEK(65314)

**Result:** Returns an even number if printer is on or an odd number if printer is off.

**Remarks:** Can be used to determine if the printer is on or off.

**Command:** PEEK(153)

**Result:** Returns comma field width for printer.

**Remarks:** None.

**Command:** PEEK(154)

**Result:** Returns last comma field width for printer.

**Remarks:** None.

**Note:** For more information on the operational aspects of your printer, consult your printer manual and pages 209-210 of 'GOING AHEAD WITH EXTENDED BASIC'.

## FOR DISK SYSTEM ONLY

**Command:** POKE 25.14:POKE 26.1:POKE 3584.0:NEW

**Result:** Makes maximum memory available for Basic programs.

**Remarks:** Clears any Basic program in memory. Does not allow the use of graphics.

**Command:** POKE 25.20:POKE 26.1:POKE 5120.0:NEW

**Result:** Sets Basic program starting address at the 'PCLEAR 1' Page.

**Remarks:** Allows limited use of graphics. Clears any Basic program in memory.

**Command:** POKE 25.26:POKE 26.1:POKE 6656.0:NEW

**Result:** Sets Basic program starting address at the 'PCLEAR 2' Page.

**Remarks:** Same as for preceding command.

**Command:** POKE 25.32:POKE 26.1:POKE 8192.0:NEW

**Result:** Sets Basic program starting address at the 'PCLEAR 3' Page.

**Remarks:** Same as for preceding command.

**Command:** POKE 25.38:POKE 26.1:POKE 9728.0:NEW

**Result:** Sets Basic program starting address at the 'PCLEAR 4' Page.

**Remarks:** PCLEAR 4 is Basic default mode.

**Command:** PEEK(234)

**Result:** Returns 2 if a sector is being/was read from disk.

**Remarks:** None.

**Command:** PEEK(234)

**Result:** Returns 3 if information is being/was written to disk.

**Remarks:** None.

**Command:** PEEK(235)

**Result:** Returns the current disk drive in operation.

**Remarks:** The number returned is between 0 and 3.

**Command:** PEEK(236)

**Result:** Returns current track number.

**Remarks:** Track number is between 0 and 34.

**Command:** PEEK(237)

**Result:** Returns current sector number.

**Remarks:** Sector number is between 1 and 18.

**Command:** PEEK(240)

**Result:** Returns 0 if no error occurred during disk I/O.

**Remarks:** Memory location 240 contains the disk status.

**Command:** PEEK(240)

**Result:** Returns 128 if disk drive is not ready during disk I/O.

**Remarks:** Also returns 128 if an ?IOERROR occurred during disk I/O.

**Command:** PEEK(240)

**Result:** Returns 64 if information was being written on a write-protected disk.

**Remarks:** None.

**Command:** POKE 308,0:POKE 313,0

**Result:** Disables all disk commands and functions.

**Remarks:** None.

**Command:** POKE 308,19:POKE 313,6

**Result:** Restores all disk commands/functions after preceding command.

**Remarks:** Only compatible with Disk Basic 1.0

**Command:** POKE 308,20:POKE 313,6

**Result:** Restores all disk commands/functions after 'POKE 308,0:POKE 313,0'.

**Remarks:** Only compatible with Disk Basic 1.1.

**Command:** FOR I = 2048 TO 2048 + 73:PRINT CHR\$(PEEK(I));:NEXT I

**Result:** Returns the file allocation table for disk drive 0.

**Remarks:** The above command must be used after the 'DIR' Command.

**Command:** FOR I = 2122 TO 2122 + 73:PRINT CHR\$(PEEK(I));:NEXT I

**Result:** Returns the file allocation table for disk drive 1.

**Remarks:** Same as for preceding command.

**Command:** FOR I = 2196 TO 2196 + 73:PRINT CHR\$(PEEK(I));:NEXT I

**Result:** Returns the file allocation table for disk drive 2.

**Remarks:** Same as for preceding command.

**Command:** FOR I = 2270 TO 2270 + 73:PRINT CHR\$(PEEK(I));:NEXT I

**Result:** Returns the file allocation table for disk drive 3.

**Remarks:** Same as for preceding command.

**Command:** FOR I = 2380 TO 2387:PRINT CHR\$(PEEK(I));:NEXT I

**Result:** Returns the filename of the most recently loaded disk file.

**Remarks:** None.

**Command:** FOR I = 2388 TO 2391:PRINT CHR\$(PEEK(I));:NEXT I

**Result:** Returns the extension of the most recently loaded disk file.

**Remarks:** None.

**Command:** PEEK(2391)

**Result:** Returns 0 if the most recent disk loaded file is Basic.

**Remarks:** The 4 types of files are: Basic, Data, ML & Text Editor Source.

**Command:** PEEK(2391)

**Result:** Returns 1 if the most recently disk loaded file is Data.

**Remarks:** None.

**Command:** PEEK(2391)

**Result:** Returns 2 if the most recently disk loaded file is ML.

**Remarks:** None.

**Command:** PEEK(2391)

**Result:** Returns 3 if the most recently disk loaded file is a Text Editor Source file.

**Remarks:** None.

**Command:** PEEK(2392)

**Result:** Returns 0 if the most recently disk loaded file has a crunched or a binary format.

**Remarks:** None.

**Command:** PEEK(2392)

**Result:** Returns 255 if the most recently disk loaded file has an ASCII format.

**Remarks:** None.

**Command:** PEEK(2394)

**Result:** Returns current disk drive in operation.

**Remarks:** Set by the 'DRIVE x' Command.

**Command:** PEEK(2395)

**Result:** Returns the number of disk file control buffers active.

**Remarks:** The no. of file control buffers are between 1 and 15.

**Command:** POKE 2396,255

**Result:** Erases any Basic program in memory when reset is pressed.

**Remarks:** Also closes all files and shuts all file control buffers.

**Command:** POKE 2396,0

**Result:** Restores to normal after 'POKE 2396,255'.

**Remarks:** None.

**Command:** POKE 2397,255:LOAD"PRONAME"

**Result:** Clears any Basic program in memory if a load error occurs.

**Remarks:** None.

**Command:** POKE 2397,0

**Result:** Restores to normal after preceding command.

**Remarks:** None.

**Command:** PEEK(2398)

**Result:** Returns 0 if no 'MERGE' Command was executed.

**Remarks:** None.

**Command:** PEEK(2398)

**Result:** Returns 255 if two Basic programs were combined by 'MERGE'.

**Remarks:** None.

**Command:** PEEK(2399)\*256 + PEEK(2400)

**Result:** Returns USR0 Vector Address.

**Remarks:** Set by 'DEFUSR0 = xxxx'.

**Command:** PEEK(2401)\*256 + PEEK(2402)

**Result:** Returns USR1 Vector Address.

**Remarks:** Set by 'DEFUSR1 = xxxx'.

**Command:** PEEK(2403)\*256 + PEEK(2404)  
**Result:** Returns USR2 Vector Address.  
**Remarks:** Set by 'DEFUSR2 = xxxx'.

**Command:** PEEK(2405)\*256 + PEEK(2406)  
**Result:** Returns USR3 Vector Address.  
**Remarks:** Set by 'DEFUSR3 = xxxx'.

**Command:** PEEK(2407)\*256 + PEEK(2408)  
**Result:** Returns USR4 Vector Address.  
**Remarks:** Set by 'DEFUSR4 = xxxx'.

**Command:** PEEK(2409)\*256 + PEEK(2410)  
**Result:** Returns USR5 Vector Address.  
**Remarks:** Set by 'DEFUSR5 = xxxx'.

**Command:** PEEK(2411)\*256 + PEEK(2412)  
**Result:** Returns USR6 Vector Address.  
**Remarks:** Set by 'DEFUSR6 = xxxx'.

**Command:** PEEK(2413)\*256 + PEEK(2414)  
**Result:** Returns USR7 Vector Address.  
**Remarks:** Set by 'DEFUSR7 = xxxx'.

**Command:** PEEK(2415)\*256 + PEEK(2416)  
**Result:** Returns USR8 Vector Address.  
**Remarks:** Set by 'DEFUSR8 = xxxx'.

**Command:** PEEK(2417)\*256 + PEEK(2418)  
**Result:** Returns USR9 Vector Address.  
**Remarks:** Set by 'DEFUSR9 = xxxx'.

**Command:** PEEK(2428) \* 256 + PEEK(2429)

**Result:** Returns current direct access file record length.

**Remarks:** Set by 'OPEN "D",#1,"DATA/DAT" xx'

**Command:** PEEK(2430)

**Result:** Returns current track number for drive 0.

**Remarks:** Track number is between 0 and 34.

**Command:** PEEK(2431)

**Result:** Returns current track number for drive 1.

**Remarks:** Same as for preceding command.

**Command:** PEEK(2432)

**Result:** Returns current track number for drive 2.

**Remarks:** Same as for preceding command.

**Command:** PEEK(2433)

**Result:** Returns current track number for drive 3.

**Remarks:** Same as for preceding command.

**Command:** POKE 2439,255

**Result:** Turns on verify flag.

**Remarks:** Same as 'VERIFY ON' command.

**Command:** POKE 2439,0

**Result:** Turns off verify flag.

**Remarks:** Same as 'VERIFY OFF' command.

**Command:** PEEK(2439)

**Result:** Returns 255 if verify flag is on.

**Remarks:** None.

**Command:** PEEK(2439)

**Result:** Returns 0 if verify flag is off.

**Remarks:** None.

**Command:** EXEC 49152

**Result:** Produces the disk copyright message.

**Remarks:** Clears any Basic program in memory.

**Command:** PEEK(49152)

**Result:** Returns 68 if Disk Basic ROM is present.

**Remarks:** Can be used to distinguish between a Disk & Cassette system.

**Command:** PEEK(49156) \* 256 + PEEK(49157)

**Result:** Returns the address of the 'DSKCON' Routine.

**Remarks:** The 'DSKCON' Routine is used for disk I/O.

**Command:** EXEC 49364

**Result:** Enters the warmstart routine.

**Remarks:** Only compatible with Disk Basic 1.0.

**Command:** EXEC 49383

**Result:** Enters the warmstart routine.

**Remarks:** Only compatible with Disk Basic 1.1.

**Command:** EXEC 52175.

**Result:** Displays the disk directory on the screen.

**Remarks:** Only compatible with Disk Basic 1.0.

**Command:** EXEC 52393

**Result:** Displays the disk directory on the screen.

**Remarks:** Only compatible with Disk Basic 1.1.

**Command:** POKE 65344.1

**Result:** Turns on the disk drive 0 light.

**Remarks:** Disk drive 0 must be on.

**Command:** POKE 65344.2

**Result:** Turns on the disk drive 1 light.

**Remarks:** Disk drive 1 must be on.

**Command:** POKE 65344.4

**Result:** Turns on disk drive 2 light.

**Remarks:** Disk drive 2 must be on.

**Command:** POKE 65344.64

**Result:** Turns on disk drive 3 light.

**Remarks:** Disk drive 3 must be on.

**Command:** POKE 65344.0

**Result:** Shuts off all disk drive motors.

**Remarks:** Can be used as — "LOADM" "MLPRO":POKE 65344.0:EXEC".

**Command:** PEEK(65353)

**Result:** Returns the current track no. of the disk drive in operation.

**Remarks:** Track no. is between 0 and 34.

**Command:** PEEK(65354)

**Result:** Returns current sector number of the disk drive in operation.

**Remarks:** Sector number is between 1 and 18.

**Command:** PEEK(65352)

**Result:** Returns 0 if no error occurred during disk I/O.

**Remarks:** None.

## DISK COMMANDS DISABLES

**Command:** A = PEEK(116) \* 256 + PEEK(117) - 196: X = INT (A / 256): Y = A - (X \* 256): B = A: FOR I = &HC17F TO &HC1DB: POKE B, PEEK(I): B = B + 1: NEXT I: POKE 309, X: POKE 310, Y: CLEAR 200, A

**Result:** Sets the base for disk commands disables. For Disk Basic 1.0.

**Remarks:** Any one or more disk commands disables may be used following the command given above.

**Command:** A = PEEK(116) \* 256 + PEEK(117) - 196: X = INT (A / 256): Y = A - (X \* 256): B = A: FOR I = &HC192 TO &HC1F1: POKE B, PEEK(I): B = B + 1: NEXT I: POKE 309, X: POKE 310, Y: CLEAR 200, A

**Result:** Sets the base for disk commands disables. For Disk Basic 1.1.

**Remarks:** Any one or more disk commands disables may be used following the command given above.

**Command:** POKE 32571, 68

**Result:** Disables 'DIR' Command for 32K/64K.

**Remarks:** Use after Disk Basic base POKES. For 16K Use 'POKE 16187, 68'.

**Command:** POKE 32571, 73

**Result:** Restores 'DIR' Command for 32K/64K.

**Remarks:** Use after Disk Basic base POKES. For 16K use 'POKE 16187, 73'.

**Command:** POKE 32589, 68

**Result:** Disables 'KILL' Command for 32K/64K.

**Remarks:** Use after Disk Basic base POKES. For 16K use 'POKE 16205, 68'.

**Command:** POKE 32589, 73

**Result:** Restores 'KILL' Command for 32K/64K.

**Remarks:** Use after Disk Basic base POKES. For 16K use 'POKE 16205, 73'.

**Command:** POKE 32593,68

**Result:** Disables 'LOAD' Command for 32K/64K.

**Remarks:** Use after Disk Basic base POKES. For 16K use 'POKE 16209,68'.

**Command:** POKE 32593,79

**Result:** Restores 'LOAD' Command for 32K/64K.

**Remarks:** Use after Disk Basic base POKES. For 16K use 'POKE 16209,79'.

**Command:** POKE 32601,68

**Result:** Disables 'MERGE' Command for 32K/64K.

**Remarks:** Use after Disk Basic base POKES. For 16K use 'POKE 16217,68'.

**Command:** POKE 32601,69

**Result:** Restores 'MERGE' Command for 32K/64K.

**Remarks:** Use after Disk Basic base POKES. For 16K use 'POKE 16217,69'.

**Command:** POKE 32606,68

**Result:** Disables 'RENAME' Command for 32K/64K.

**Remarks:** Use after Disk Basic base POKES. For 16K use 'POKE 16222,68'.

**Command:** POKE 32606,69

**Result:** Restores 'RENAME' Command for 32K/64K.

**Remarks:** Use after Disk Basic base POKES. For 16K use 'POKE 16222,69'.

**Command:** POKE 32616,68

**Result:** Disables 'SAVE' Command for 32K/64K.

**Remarks:** Use after Disk Basic base POKES. For 16K use 'POKE 16232,68'.

**Command:** POKE 32616,65

**Result:** Restores 'SAVE' Command for 32K/64K.

**Remarks:** Use after Disk Basic base POKES. For 16K use 'POKE 16232,65'.

**Command:** POKE 32637,68

**Result:** Disables 'DSKINI' Command for 32K/64K.

**Remarks:** Use after Disk Basic base POKES. For 16K use 'POKE 16253,68'.

**Command:** POKE 32637,83

**Result:** Restores 'DSKINI' Command for 32K/64K.

**Remarks:** Use after Disk Basic base POKES. For 16K use 'POKE 16253,83'.

**Command:** POKE 32643,68

**Result:** Disables 'BACKUP' Command for 32K/64K.

**Remarks:** Use after Disk Basic base POKES. For 16K Use 'POKE 16259,68'.

**Command:** POKE 32643,65

**Result:** Restores 'BACKUP' Command for 32K/64K.

**Remarks:** Use after Disk Basic base POKES. For 16K use 'POKE 16259,65'.

**Command:** POKE 32653,68

**Result:** Disables 'DSKIS' Command for 32K/64K.

**Remarks:** Use after Disk Basic base POKES. For 16K use 'POKE 16269,68'.

**Command:** POKE 32653,83

**Result:** Restores 'DSKIS' Command for 32K/64K.

**Remarks:** Use after Disk Basic base POKES. For 16K use 'POKE 16269,83'.

**Command:** POKE 32658,68

**Result:** Disables 'DSKOS' Command for 32K/64K.

**Remarks:** Use after Disk Basic base POKES. For 16K use 'POKE 16274,68'.

**Command:** POKE 32658,83

**Result:** Restores 'DSKOS' Command for 32K/64K.

**Remarks:** Use after Disk Basic base POKES. For 16K use 'POKE 16274,83'.

**Note:** Once you have typed in the BASE POKES for Disk Basic Command Disables, you may use ONE or MORE Disk Basic Command Disable POKES.

## FOR 64K CASSETTE & DISK SYSTEMS

**Command:** RESTORE: FOR I = 3584 TO 3584 + 24: READ A:POKE I,A: NEXT: EXEC 3584: DATA 26,80,142,128,0,166, 132,183,255,223,167,128,140,224,0,39,5,183, 255,222,32,239,28,175,57

**Result:** Sets the 64K ALL RAM MODE.

**Remarks:** Resetting the computer will deactivate this mode.

**Note:** All commands listed under this section require the ALL RAM MODE.

**Command:** POKE 41165,0:EXEC 41085

**Result:** Sets the computer to Color Basic.

**Remarks:** Disables ECB and Disk Basic.

**Command:** POKE 41381,18

**Result:** Generates a steady cursor.

**Remarks:** None.

**Command:** POKE 41382,188

**Result:** Generates a blinking cursor.

**Remarks:** 'POKE' different values to create various cursor types.

**Command:** POKE 41385,18:POKE 41386,18

**Result:** Generates an invisible cursor.

**Remarks:** None.

**Command:** POKE 41593,0

**Result:** Does not allow uppercase/lowercase flip thru 'SHIFT-0'.

**Remarks:** None.

**Command:** POKE 41593.18

**Result:** Restores to normal after 'POKE 41593.0'.

**Remarks:** Allows uppercase/lowercase flip thru 'SHIFT-0'.

**Command:** POKE 41893.0

**Result:** Disables the Clear Key on INPUT/LINEINPUT commands.

**Remarks:** None.

**Command:** POKE 41893.129

**Result:** Restores Clear Key Operation to normal after preceding POKE.

**Remarks:** None.

**Command:** POKE 41923.0

**Result:** Disables the Break Key on INPUT/LINEINPUT commands.

**Remarks:** None.

**Command:** POKE 41923.3

**Result:** Restores Break Key Operation after preceding POKE.

**Remarks:** None.

**Command:** POKE 41909.0

**Result:** Disables 'SHIFT-BACKSPACE' on INPUT/LINEINPUT commands.

**Remarks:** None.

**Command:** POKE 41909.21

**Result:** Restores 'SHIFT-BACKSPACE' Operation after preceding POKE.

**Remarks:** None.

**Command:** POKE 41954.255

**Result:** Allows you to enter 255 characters in a single statement.

**Remarks:** Allows you to enter more instructions in a single statement.

**Command:** FOR I = 42223 TO 42231:READA:POKEI,A:NEXTI:DATA  
189, 173,33,189,172,239,126,173,158

**Result:** Autostarts your basic program from cassette by  
'CLOAD'.

**Remarks:** None.

**Command:** FOR I = 43288 TO 43302:POKE I,18:NEXT I

**Result:** Allows you 255 different designs thru 'CLS xx'.

**Remarks:** xx = 1 to 255.

**Command:** POKE 43314,5:POKE 43315,0

**Result:** 'CLS xx' clears only top half of the text screen.

**Remarks:** None.

**Command:** POKE 43307,5:POKE 43308,0

**Result:** 'CLS xx' clears only bottom half of the text screen.

**Remarks:** None.

**Command:** POKE 44009,ASC("A"):POKE 44010,ASC("T")

**Result:** Changes 'BREAK IN xx' Prompt to 'BREAK AT xx'.

**Remarks:** None.

**Command:** POKE 44014,43:POKE 44015,43

**Result:** Changes the 'OK' Prompt to A '+' + '.

**Remarks:** 'POKE' different values to create different prompts.

**Command:** FOR I = 44150 TO 44155:POKE I,18:NEXT I

**Result:** Disables the 'OK' Prompt.

**Remarks:** None.

**Command:** POKE 44451,18:POKE 44452,18

**Result:** Disables the Break Key on all but INPUT/LINEINPUT  
commands.

**Remarks:** For Color Basic only.

**Command:** POKE 44533.12

**Result:** Pauses the execution of the Basic Program when the Clear-Key is pressed.

**Remarks:** Press any key to resume the execution of your Basic program.

**Command:** POKE 45288,0

**Result:** Disables the '?EXTRA IGNORED' Message.

**Remarks:** None.

**Command:** POKE 45007,0

**Result:** Disables the '?REDO' Message.

**Remarks:** None.

**Command:** POKE 410,57:FOR I = &H82BD TO &H82BF: POKE I,18: NEXT I: POKE 411,&H82: POKE 412,&HB9: POKE 410,126

**Result:** Disables the Break Key on all but INPUT/LINEINPUT Commands.

**Remarks:** For ECB and Disk Basic only.

**Command:** POKE &H82E3,42:POKE &H82ED,42

**Result:** Changes the ECB TRON prompt from '[xx]' to '\* xx \*'.

**Remarks:** Requires ECB.

**Command:** POKE &H80A7,0:EXEC &HA07D

**Result:** Sets the computer to Extended Basic.

**Remarks:** Requires ECB. Disables Disk Basic if present.

## ROMPAK TRANSFER

Type in PROGRAM LISTING 1 and CSAVE it to a blank cassette. Turn off the computer. Remove the disk controller if you have one. Now place the ROMPAK top down with the slide cover away from you. Pull the slide cover towards you. You should see a line of "silver pins". Cover the LEFTMOST and ONLY the LEFT-MOST pin with a small piece of scotch tape. Insert the ROMPAK in the ROMPAK SLOT and turn on the computer. The normal 'EXTENDED BASIC ..' should appear. CLOAD Program Listing 1 from cassette and RUN it. Enter option 1. Press PLAY and RECORD on your cassette recorder and enter the appropriate filename when the computer asks you. The ROMPAK will be saved immediately following Program Listing 1 on cassette.

TO LOAD ROMPAK FROM CASSETTE: Turn off computer and remove ROMPAK. Do NOT plug in the disk controller if you have one. Turn on the computer, CLOAD Program Listing from cassette, RUN it, enter option 2 and enter the filename of the ROMPAK. The program will load the ROMPAK and EXECute it. Note: Send a SASE to Microcom Software for information on most problem ROMPAKS.

## PROGRAM LISTING 1

```
10 CLEAR 200,16384:CLS:PRINT"(1) SAVE ROMPAK":PRINT"(2) LOAD ROMPAK"
20 EXEC 44539:IF PEEK(135)=49 THEN 30 ELSE 110
30 PRINT:PRINT"SAVE ROMPAK TO TAPE":PRINT"SEARCHING ROMPAK .....";:X=(PEEK(136)*256+PEEK(137))-1024
40 A=PEEK(49152):B=PEEK(49153):C=PEEK(49154):D=PEEK(49155):E=PEEK(49156):F=PEEK(49157)
50 FOR I=$HC2000 TO $HFE00 STEP $H100:PRINT @X,HEX$(I):IF PEEK(I)=A AND PEEK(I+1)=B AND PEEK(I+2)=C AND PEEK(I+3)=D AND PEEK(I+4)=E AND PEEK(I+5)=F THEN 80
60 IF PEEK(I)=126 AND PEEK(I+1)=126 AND PEEK(I+2)=126 AND PEEK(I+3)=126 AND PEEK(I+4)=126 AND PEEK(I+5)=126 THEN 80
70 NEXT
80 IF I=$HFF000 THEN I=$HFEFFF
90 INPUT"FILENAME":;FI$:CSAVEM FI$,49152,I,49152
100 GOTO 10
110 RESTORE:FOR I=3584 TO 3634:READ A:POKE I,A:NEXT
120 DATA 26, 80, 142, 128, 0, 183, 255, 222, 166, 132, 183, 255, 223,
167, 128, 140, 224, 0, 35, 241, 142, 14, 34, 191, 128, 190, 15, 113,
127, 192, 0, 126, 168, 161, 142, 192, 0, 206, 64, 0
130 DATA 166, 192, 167, 128, 156, 243, 35, 248, 126, 192, 0
140 INPUT"FILENAME":FI$:CLOADM FI$,&H8000
150 A=PEEK(126)*256+PEEK(127)-1:A=A+32768:A$=HEX$(A):POKE &HF3,VAL("&H"+LEFT$(A$,2)):POKE &HF4,VAL("&H"+RIGHT$(A$,2))
160 PRINT"ROMPAK READY TO EXECUTE":EXEC 42961:EXEC 3584
```

## GAMES

**Note:** Type in these POKEs AFTER '(C)LOADM'ing your arcade-game and BEFORE typing 'EXEC'.

**Command:** POKE 6415,x

**Result:** Sets 'x' ships in Astro Blast.

**Remarks:** x is any number of ships you want from 1 to 255.

**Command:** POKE 5872,x

**Result:** Sets 'x' Men in Berserk.

**Remarks:** x is any number of men you want from 1 to 255.

**Command:** POKE 15121,x

**Result:** Sets 'x' Men in Dunkey Munkey.

**Remarks:** x is any number of men you want from 1 to 255.

**Command:** POKE 4329,x

**Result:** Sets 'x' lives in Doubleback.

**Remarks:** x is any number of lives you want from 1 to 255.

**Command:** POKE 12914,x

**Result:** Sets 'x' Men in Donkey King.

**Remarks:** x is any number of men you want from 1 to 255.

**Command:** POKE 12904,x

**Result:** Sets 'x' Men in Donkey King Practice Game.

**Remarks:** x is any number of men you want from 1 to 200.

**Command:** POKE 12889,2

**Result:** Starts the Donkey King Game at the Elevators Level.

**Remarks:** None.

**Command:** POKE 12889,10

**Result:** Starts the Donkey King Game at the Rivits Level.

**Remarks:** None.

**Command:** POKE 12889,18

**Result:** Starts the Donkey King Game at the Conveyor Belts Level.

**Remarks:** None.

**Command:** POKE 14019,x

**Result:** Sets 'x' lives in Electron.

**Remarks:** x is any number of lives you want from 1 to 255.

**Command:** POKE 10739,x

**Result:** Sets 'x' lives in Katerpillar.

**Remarks:** x is any number of lives you want from 1 to 255.

**Command:** POKE 21115,x

**Result:** Sets 'x' lives in Ms. Ghost.

**Remarks:** x is any number of lives you want from 1 to 255.

**Command:** POKE 9009,x

**Result:** Sets 'x' lives in Pac—Tac.

**Remarks:** x is any number of lives you want from 1 to 255.

**Command:** POKE 10489,x

**Result:** Sets 'x' lives in Pac—Tac 2

**Remarks:** x is any number of lives you want from 1 to 255.

**Command:** POKE 13579.90

**Result:** The same joystick controls you use for vertical movement in 'PROTECTORS' also control the way you shoot.

**Remarks:** None.

**Command:** POKE 7674,x

**Result:** Sets 'x' Ships in Space Invaders.

**Remarks:** x is any number of ships you want from 1 to 255.

**Command:** POKE 10638,x

**Result:** Sets 'x' Ships in Space Race.

**Remarks:** x is any number of ships you want from 1 to 255.

**Command:** POKE 12002,x

**Result:** Sets 'x' Lives in 'THE FROG'.

**Remarks:** x is any number of lives you want from 1 to 255.

**Command:** POKE 13951,x

**Result:** Sets 'x' Ships in Fury.

**Remarks:** x is any number of ships you want from 1 to 255.

**Command:** POKE 9075,x

**Result:** Sets 'x' Lives in Ghost Gobbler.

**Remarks:** x is any number of lives you want from 1 to 255.

**Command:** POKE 2414,x

**Result:** Sets 'x' Men in Haunted Treasure.

**Remarks:** x = 1 to 255. Type in the above POKE \* Before \* Loading program.

**Command:** POKE 7446,x

**Result:** Sets 'x' Ships in Invasion.

**Remarks:** x is any number of ships you want from 1 to 40.

**Command:** POKE 11452,x

**Result:** Sets 'x' Men in Trapfall.

**Remarks:** x is any number of men you want from 1 to 255.

**Command:** POKE 8312,x

**Result:** Sets 'x' Ships in Whirlybird Run.

**Remarks:** x is any number of ships you want from 1 to 255.

**Command:** POKE 16670,x

**Result:** Sets 'x' Ships in Zaksund.

**Remarks:** x is any number of ships you want from 1 to 255.

**Command:** POKE 25624,x

**Result:** Sets 'x' Ships in Zaxxon. For Disk version only.

**Remarks:** x is any number of ships you want from 1 to 255.

## More Lives On 'CANYON CLIMBER'

**Step 1:** Turn on Computer.

**Step 2:** Type: POKE 65315,54 and Press [Enter].

**Step 3:** Insert 'CANYON CLIMBER' Cartridge in Cartridge Slot.

**Step 4:** Type: EXEC 49222 and press [Enter].

**Step 5:** Push the reset button.

## WHAT THIS BOOK CAN DO FOR YOU

This book will help you gain the power of Assembly Language thru Basic and 'get underneath the cover' of the Color Computer. It will also help you to utilize the System Configuration and will offer you the following fine features:

- Enhance the capability of your Basic programs.
- Make possible various cassette, disk and printer functions
- Fully utilize the potential of the Basic language
- Fully compatible with 16K / 32K / 64K Color Basic / ECB Cassette and Disk Systems
- Fully compatible with CoCo I and CoCo II

This book includes, inter alia, POKEs, PEEKs, and EXECs to:

- Autostart your Basic programs.
- Disable Color Basic commands like LIST, LLIST, POKE, EXEC, CSAVE, CSAVE, CLOAD, CLOADM, and NEW.
- Disable ECB commands like DEL, EDIT, TRON, TROFF, PCLEAR, DLOAD, RENUM, and PRINT USING
- Disable Disk Basic Commands like DIR, KILL, SAVE, LOAD, MERGE, RENAME, DSKINI, BACKUP, DSKIS, and DSKOS
- Disable BREAK KEY, CLEAR KEY, and RESET BUTTON.
- Generate a Repeat-Key.
- Transfer ROMPAKS to tape (For 64K only).
- Speed up your Basic programs
- RESET, MOTOR ON / OFF from keyboard.
- Restart your Basic program thru RESET button.
- Produce Key-Clicks and Error-Beeps.
- Recover Basic programs lost by NEW, ?I / O ERRORS and faulty RESET.
- Set 23 different GRAPHIC / SEMIGRAPHIC modes
- Set 15 of the most commonly used Printer Baud Rates.
- Allow you more plays in 23 of your favorite arcade games.
- AND MUCH MUCH MORE !!!



**MICROCOM SOFTWARE**  
PO BOX 214, FAIRPORT, NY 14450  
PH: (716) 223-1477